



Solução de Inteligência Empresarial na Área da Energia

Inês Marina Gouveia Roque

Mestrado em Engenharia Informática
Especialização em Sistemas de Informação

Trabalho de Projeto orientado por:
António Manuel Silva Ferreira
Nuno Miguel Pereira Gomes

Agradecimentos

Aos meus pais, que sempre fizeram (e fazem) tudo por mim e pela minha irmã, com amor e dedicação, para que tenhamos o melhor futuro possível.

À minha irmã, que durante todo este percurso acreditou nas minhas capacidades e me motivou a ser e a fazer melhor.

Ao meu primo Tiago, por ser um entusiasta do meu trabalho, pelas aventuras partilhadas, por animar as minhas férias, por ser mais irmão que primo.

Aos meus tios, primos, avós e restantes familiares, que me incentivaram e apoiaram ao longo da minha vida académica.

À Vânia, ao João e à Débora, pelos almoços/lanches/jantares/cafés, pelas conversas infinitas, por me ouvirem e me aconselharem, por ficarem genuinamente felizes com as minhas pequenas conquistas, por serem meus amigos.

Ao Rui, por me incentivar a escrever quando não me apetecia, pelos conselhos, por me apoiar, pelo carinho e por me fazer sorrir.

A todos os elementos da equipa do SGC (aos que estavam no início e aos que estão hoje), bem como aos elementos das equipas de *Analytics* e *Information Technology/Operational Technology*, por me terem integrado tão bem na equipa ECS e na empresa, por responderem às minhas perguntas chatas, por partilharem o conhecimento deles comigo, pelos almoços na COPA e pela boa disposição (nem sempre, mas a maior parte das vezes) no local de trabalho.

Aos meus orientadores. Ao professor António Ferreira, pela preciosa orientação que me proporcionou, especialmente na escrita do relatório, e ao Nuno Gomes pelo apoio essencial na introdução ao mundo do OBIEE e dos *dashboards*, pela preocupação com o meu trabalho, com os prazos e com o relatório.

Não posso terminar sem lembrar o meu tio e a minha avó que, muito embora a sua partida tenha dificultado estes últimos dois anos, sei que me guiaram e me deram forças todos os dias para terminar esta importante etapa da minha vida.

A todos, muito obrigada.

Aos meus pais e irmã.

Resumo

O presente relatório descreve o trabalho realizado ao longo de um estágio na empresa Novabase, entre setembro de 2015 e junho de 2016, no âmbito do projeto “Solução de Inteligência Empresarial na Área da Energia” cujos principais objetivos foram o desenvolvimento de um processo de *Extraction-Transformation-Loading* (ETL) ágil e eficiente e a concretização dos relatórios operacionais do Sistema de Gestão de Clientes (SGC), bem como de um conjunto de *dashboards*.

Inicialmente, analisaram-se o processo de ETL e as bases de dados existentes no SGC. Uma vez que não havia qualquer modelo do sistema, concretizou-se um modelo relacional que apresenta os principais conceitos envolvidos bem como as relações entre os mesmos, permitindo obter uma visão abrangente e aprofundada dos dados.

Depois, desenvolveu-se um processo de ETL composto por múltiplos *jobs*. Após a constatação do seu elevado tempo de execução, procedeu-se à sua otimização com base na estratégia de carregamento de dados em massa, tendo permitido a melhoria dos tempos de execução dos *jobs*. A avaliação deste processo passou pela análise da conformidade dos dados obtidos face aos dados do SGC e pela verificação dos seus tempos de execução.

Seguidamente, implementou-se um conjunto de relatórios operacionais, os quais foram avaliados segundo os requisitos do cliente, as funcionalidades expectáveis e a interface com o utilizador.

Por fim, surgiu ainda a hipótese de construir alguns *dashboards* de BI, no âmbito de duas provas de conceito. Estes *dashboards* foram avaliados tendo em conta a informação que disponibilizam e a forma como esta se apresenta ao utilizador.

O estágio permitiu aplicar e aprofundar os conhecimentos adquiridos ao longo do percurso académico bem como melhorar a capacidade de trabalho em equipa, gestão de tarefas e a compreensão do mundo empresarial.

Palavras-chave: *Business Intelligence*, Processo de ETL, Relatórios Operacionais, *Dashboards*.

Abstract

This report describes the work done during an internship at the company Novabase, between September 2015 and June 2016, under the project “Solução de Inteligência Empresarial na Área da Energia” whose main goals were the development of an agile and efficient Extraction-Transformation-Loading (ETL) process and the creation of operational reports for Sistema de Gestão de Clientes (SGC), as well as a set of dashboards.

Initially, the ETL process and the existing databases were analyzed. Since there was no system data model, it was designed a relational model that presents the key concepts involved and the relationships between them, allowing a deep and comprehensive view of the data.

Then, an ETL process consisting of multiple jobs was developed. After finding the high execution time required by the jobs, it proceeded to its optimization based on bulk data loading strategy that allowed an improvement in the jobs’ runtimes. The evaluation of this process includes the analysis of data compliance between the SGC data and the obtained data and the verification of its execution times.

Next, a set of operational reports were implemented, which were evaluated according to customer requirements, the expected functionality and user interface.

Finally, it was possible to build some BI dashboards, within two proves of concept. These dashboards were evaluated taking into account the information they provide and how this information is presented to the user

This internship allowed to apply and deepen the knowledge acquired during the academic course and to improve the ability to work in teams, task management and understanding the business world.

Keywords: *Business Intelligence, ETL Process, Operational Reports, Dashboards*

Índice

Lista de figuras.....	ix
Lista de tabelas.....	xi
Lista de acrónimos	xiii
Capítulo 1 Introdução.....	1
1.1 Motivação	1
1.2 Objetivos.....	2
1.3 Instituição de acolhimento.....	3
1.4 Planeamento e execução	4
1.5 Contribuições.....	5
1.6 Notação adotada	5
1.7 Estrutura do documento.....	6
Capítulo 2 Conceitos	7
2.1 <i>Business Intelligence</i>	7
2.2 Processo de ETL.....	9
2.3 Sistemas ERP.....	11
2.4 Relatórios de apoio à decisão	12
2.5 Sumário.....	13
Capítulo 3 Trabalho realizado	15
3.1 Ambiente de trabalho.....	15
3.1.1 Ferramentas utilizadas.....	15
3.1.2 Organização do trabalho na equipa	20
3.1.3 Processo de desenvolvimento de <i>software</i>	21

3.2	Caracterização dos dados dos processos	23
3.2.1	Contratação e serviço ao cliente.....	25
3.2.2	Gestão de equipamentos.....	26
3.2.3	Gestão de leituras	27
3.2.4	Faturação e cobranças	28
3.2.5	Gestão de trabalho.....	29
3.3	Desenvolvimento do processo de ETL.....	30
3.3.1	Preparação do ambiente de desenvolvimento	32
3.3.2	Parametrização de campos SAP.....	33
3.3.3	Extração de dados.....	34
3.3.4	Transformação de dados.....	36
3.3.5	Monitorização de erros.....	41
3.3.6	Otimização do desempenho	41
3.3.7	Validação do trabalho realizado	45
3.4	Relatórios operacionais.....	45
3.4.1	Análise de requisitos	46
3.4.2	Implementação dos relatórios.....	47
3.4.3	Problema de usabilidade e sua resolução	49
3.4.4	Validação do trabalho realizado	50
3.5	<i>Dashboards</i> de BI.....	51
3.5.1	Análise preliminar	51
3.5.2	Construção dos <i>dashboards</i>	52
3.5.3	Validação do trabalho realizado	57
3.6	Sumário.....	57
Capítulo 4	Conclusão	59
4.1	Principais contribuições.....	59
4.2	Competências adquiridas	60
4.3	Principais dificuldades.....	60
4.4	Trabalho futuro	61

Bibliografia	63
Apêndices	67
Apêndice I: Organização da equipa do SGC	69
Apêndice II: SQL dos principais objetos da base de dados	71
Apêndice III: SQL das funções de monitorização de erros	77

Lista de figuras

Figura 1.1 Organograma da Novabase e unidade de negócio onde se desenvolveu o trabalho deste PEI (fundo azul)	3
Figura 1.2 Planeamento do PEI	5
Figura 2.1 Arquitetura de uma ferramenta de BI	8
Figura 2.2 Processo de ETL	10
Figura 3.1 Interface de utilização do <i>SQL Power Architect</i>	16
Figura 3.2 Interface de utilização do Talend	17
Figura 3.3 Interface de utilização do Pentaho	19
Figura 3.4 Interface de utilização do OBIEE (edição de <i>dashboard</i>)	20
Figura 3.5 Esquema de funcionamento da equipa	21
Figura 3.6 Fluxo do processo de desenvolvimento de <i>software</i> Scrum	22
Figura 3.7 Modelo de dados simplificado do SGC	25
Figura 3.8 Modelo de dados detalhado relativo ao processo de negócio da Contratação e Serviço ao Cliente	26
Figura 3.9 Modelo de dados detalhado relativo ao processo de negócio da Gestão de Equipamentos	27
Figura 3.10 Modelo de dados detalhado relativo ao processo de negócio da Gestão de Leituras	28
Figura 3.11 Modelo de dados detalhado relativo ao processo de negócio da Faturação e Cobranças	29
Figura 3.12 Modelo de dados detalhado relativo ao processo de negócio da Gestão de Trabalho	30
Figura 3.13 Fluxo de dados do processo de ETL	31
Figura 3.14 Menu de definição do contexto do Talend	33
Figura 3.15 <i>Job</i> de extração de dados de um domínio	36

Figura 3.16 <i>Job</i> de transformação de dados do domínio Tipo de Cliente	37
Figura 3.17 <i>Job</i> de transformação de múltiplos domínios (Classe de Contacto e Subclasse de Contacto)	38
Figura 3.18 Mapeamento de campos na fase I de transformação	39
Figura 3.19 Mapeamento de campos na fase II de transformação	40
Figura 3.20 <i>Subjob</i> de controlo de erros	41
Figura 3.21 <i>Job</i> de extração de dados modificado	43
Figura 3.22 Visão geral do processo de ETL	45
Figura 3.23 Exemplo de um relatório operacional (Planos de Pagamento)	46
Figura 3.24 Elementos de um relatório operacional (Relatório de Dívidas)	46
Figura 3.25 Interface de edição de <i>parameters</i> no Pentaho	48
Figura 3.26 Interface de definição de <i>queries</i> do Pentaho (em cima) e detalhe da condição WHERE (em baixo)	49
Figura 3.27 Etapas do processo de desenvolvimento dos <i>dashboards</i>	51
Figura 3.28 Interface de edição de uma análise do OBIEE (separador Critérios) ..	53
Figura 3.29 Definição do tipo de visualização – gráfico <i>donut</i>	54
Figura 3.30 Interface de edição de um <i>dashboard</i> do OBIEE	55
Figura 3.31 <i>Dashboard</i> relativo a <i>Strategic Indicators</i>	55
Figura 3.32 <i>Dashboard</i> relativo a <i>Operations</i>	56
Figura 3.33 <i>Dashboard</i> relativo a <i>Assets</i>	56

Lista de tabelas

Tabela 3.1 Dados relativos à execução dos <i>jobs</i> de extração.....	42
Tabela 3.2 Resultados obtidos após a modificação dos <i>jobs</i> de extração	43
Tabela 3.3 Dados relativos à execução dos <i>jobs</i> de transformação	44
Tabela 3.4 Resultados obtidos após a modificação dos <i>jobs</i> de transformação.....	44

Lista de acrónimos

Acrónimo	Significado
ABAP	<i>Advanced Business Application Programming</i>
BI	<i>Business Intelligence</i>
BBP	<i>Business BluePrint</i>
DW	<i>Data Warehouse</i>
EIS	<i>Executive Information Systems</i>
ERP	<i>Enterprise Resource Planning</i>
ETL	<i>Extraction-Transformation-Loading</i>
KPI	<i>Key Performance Indicator</i>
PEI	Projeto de Engenharia Informática
PL/pgSQL	<i>Procedural Language/PostgreSQL</i>
RFC	<i>Remote Function Call</i>
SAP	<i>Systems, Applications and Products</i>
SAP IS-U	<i>SAP Industry Solution for Utilities</i>
SGC	Sistema de Gestão de Clientes
SMAS	Serviços Municipalizados de Água e Saneamento
SQL	<i>Structured Query Language</i>
UFE	<i>Unified Front-End</i>
XML	<i>eXtensible Markup Language</i>

Capítulo 1

Introdução

O Projeto de Engenharia Informática (PEI) a que se refere o presente documento esteve inserido no âmbito do *Business Intelligence* (BI) e decorreu em contexto empresarial, nomeadamente na empresa Novabase. Neste capítulo, apresenta-se a motivação para o estágio, descrevem-se os objetivos definidos, a instituição de acolhimento, o plano de trabalho e sua execução, as principais contribuições do projeto, a notação utilizada e organização do documento.

1.1 Motivação

Diariamente são geradas enormes quantidades de dados por diversos dispositivos, como os *smartphones*, *tablets*, sensores, contadores de eletricidade inteligentes e outros dispositivos com ligação à internet, que se juntam à significativa quantidade de informação gerada pelos meios tradicionais (televisão, *web*, livros) [1].

Com este aumento de dados que tem surgido nos últimos anos, conceitos como *customer insight* (“visão do cliente”) e *analytics* têm vindo a ganhar terreno em várias áreas de negócio. As empresas reconhecem o vasto número de oportunidades disponibilizado por este elevado volume de dados, nomeadamente no que concerne ao aumento de vendas e à condução da fidelidade dos clientes. No entanto, é necessário que a informação seja apresentada da forma adequada tanto para a empresa como para o cliente [1] [2].

A compreensão do comportamento, motivação e satisfação dos clientes, bem como o que os influencia, tem sido, desde sempre, uma preocupação das empresas em diferentes áreas de negócio. O conhecimento destes elementos permite às empresas adquirir e preservar clientes, bem como o desenvolvimento de novos produtos [3].

Para dar resposta aos desafios nesta área, as empresas têm dedicado grandes esforços e recursos ao desenvolvimento de processos eficientes de transformação e análise de grandes volumes de dados, ao mesmo tempo que procuram compreender

melhor os seus clientes, oferecer serviços/produtos personalizados e estar à frente da concorrência [4].

O BI não é novidade, mas nos últimos anos tem alcançado uma maior divulgação e aceitação junto das empresas. O recurso a práticas de BI subentende que existem dados provenientes da atividade empresarial passíveis de serem processados de forma a obter informação e conhecimento capazes de apoiar a tomada de decisão por parte de gestores e analistas de negócio.

Com este PEI, pretendeu-se explorar vários elementos que fazem parte do BI. No âmbito da área da Energia, o Sistema de Gestão de Clientes (SGC) incluía um conjunto de relatórios operacionais que pretendiam dar resposta a algumas necessidades do cliente, mas que não refletiam o estado atual do sistema.

1.2 Objetivos

Os principais objetivos definidos para o presente PEI foram:

Objetivo 1 (O1): Desenvolvimento de um processo de *Extraction-Transformation-Loading* (ETL) de dados armazenados num sistema *Enterprise Resource Planning* (ERP), eficiente e ágil que resulte em dados de qualidade e em conformidade com o modelo de dados do SGC. A avaliação deste objetivo consistiu na análise comparativa dos dados obtidos com os dados existentes no ERP (cuja obtenção está a cargo do processo de ETL) e na verificação dos tempos de execução do processo.

Objetivo 2 (O2): Concretização de relatórios operacionais capazes de fornecer informação operacional detalhada, bem como um conjunto de filtros, que suporte os processos de negócio e reflita o estado atual do SGC. A avaliação deste objetivo foi concretizada por elementos da equipa funcional e do cliente do SGC, que analisaram os relatórios desenvolvidos face aos requisitos definidos, ao seu funcionamento e interface com o utilizador. Esta validação dos relatórios foi mediada por uma ferramenta de rastreamento e reportação de *bugs*, que permite que seja fornecido *feedback* à distância (no caso de equipas de testes não presentes no mesmo ambiente físico) e que, uma vez que o problema fica registado, pode ser resolvido mais tarde.

Objetivo 3 (O3): Realização de *dashboards* de BI que permitam a visualização gráfica dos dados atuais e das tendências do negócio, por forma a simplificar a tomada de decisões. Este objetivo foi avaliado por elementos da equipa de *analytics*, no âmbito de duas provas de conceito, tendo em conta os objetivos das mesmas. A avaliação destes *dashboards* passou pela análise da informação disponibilizada pelos mesmos, bem como da forma como esta é apresentada.

1.3 Instituição de acolhimento

O PEI decorreu na empresa Novabase SGPS S.A., que tem escritórios em Portugal, Espanha, Reino Unido, Emiratos Árabes Unidos, Angola, Moçambique e Turquia e tem projetos em 35 países. A empresa está segmentada em áreas de negócio (*Business Solutions, Infrastructures and Managed Services* e *IT Contracting*) que dão resposta às várias indústrias: Energia, Governo, Serviços Financeiros, Telecomunicações e Transportes. O presente PEI esteve inserido na unidade de negócio *Energy Core Systems*, que está incluída no mercado de Governo, Transportes e Energia da área de negócio *Business Solutions*, tal como ilustrado pela Figura 1.1, e que tem como objetivo a entrega de soluções no sector da energia (eletricidade, água e gás).

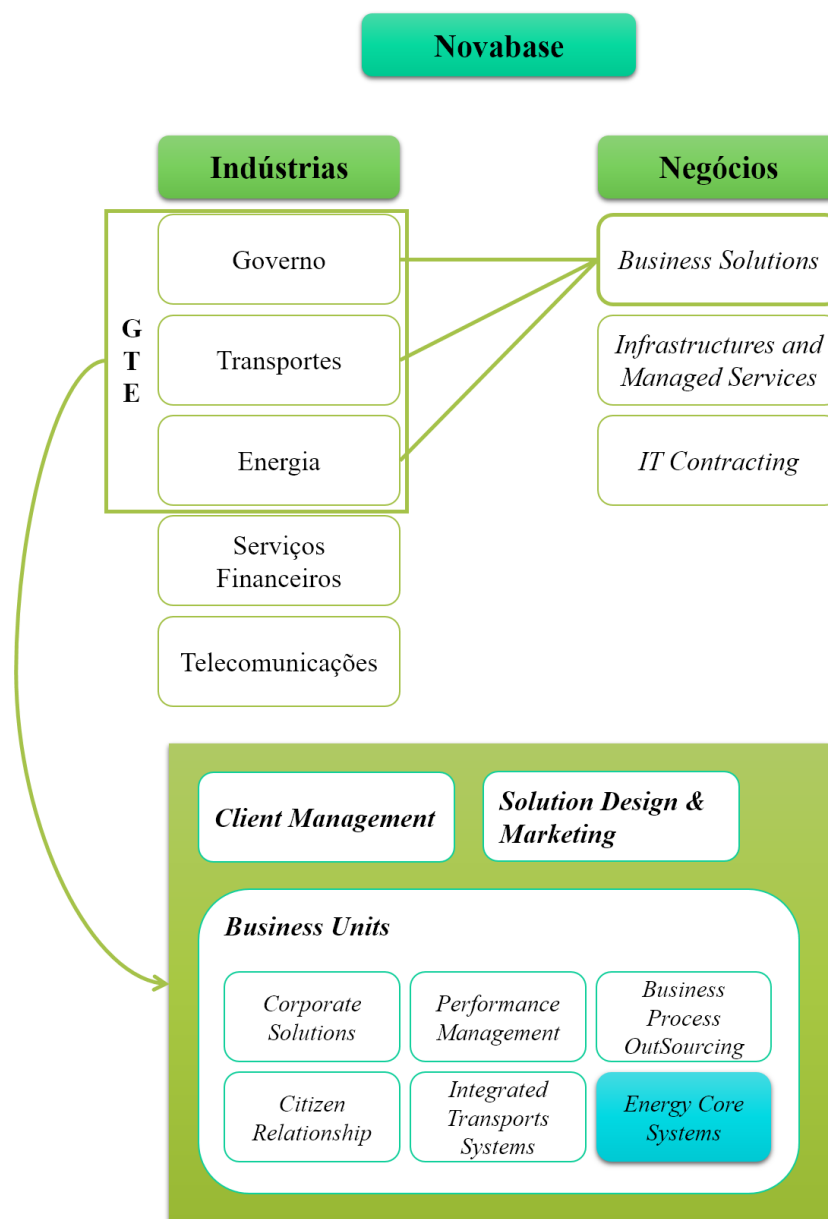


Figura 1.1 Organograma da Novabase e unidade de negócio onde se desenvolveu o trabalho deste PEI (fundo azul)

Os relatórios operacionais concretizados encontram-se inseridos no âmbito do projeto SGC. O SGC é composto por um conjunto de processos que estão a ser desenvolvidos no sistema *SAP Industry Solution for Utilities* (SAP IS-U) e por uma interface, o *Unified Front-End* (UFE), que visa facilitar a utilização do SGC por parte dos seus utilizadores.

Os processos de negócio que integram o SGC são [5]:

- **Contratação e serviço ao cliente:** processos que têm como objetivo a definição das condições entre o cliente e a empresa, nomeadamente através da concretização de um contrato de prestação de serviços;
- **Gestão de leituras e contadores:** conjunto de processos cujo objetivo é garantir o planeamento e operacionalidade das leituras (periódicas, não periódicas e comunicadas pelo cliente), bem como a manutenção dos contadores;
- **Cálculo de consumos/faturação:** contempla processos que têm por objetivo o cálculo dos consumos registados ou estimados num contrato, faturação, e impressão e entrega da fatura ao cliente para posterior cobrança;
- **Gestão de trabalho:** grupo de processos cujo objetivo é garantir a resposta a solicitações do cliente passíveis de resolução pelas equipas de atendimento/*backoffice*, tais como pedidos do cliente para concretização de intervenções no terreno;
- **Cobranças e pagamentos:** processos que têm como objetivo garantir que os pagamentos referentes à prestação de serviços por parte da empresa decorrem sem prejuízo para ambas as partes, bem como o tratamento de situações de dívida do cliente.

Foram desenvolvidos relatórios operacionais para *todos* os processos apresentados, os quais se detalham no Capítulo 3.

1.4 Planeamento e execução

Apresenta-se na Figura 1.2 o planeamento das tarefas concretizadas no PEI. Este plano está organizado segundo as principais tarefas e respetivas sub-tarefas, cuja duração se detalha através de barras horizontais demarcadoras de intervalos de tempo.

Face ao planeado, a principal alteração reside na entrega do Relatório Preliminar, o qual, como se pode constatar pela figura, não aconteceu. O principal motivo para a tomada desta decisão foi a carga de trabalho do período correspondente à entrega (entre novembro e dezembro), uma vez que, em simultâneo com o PEI (que decorreu em regime de oito horas diárias, quatro dias por semana), frequentava uma cadeira curricular que incluía trabalhos práticos e projetos. Também durante esse período decorreu uma formação da equipa do SGC para o cliente, em que alguns dos relatórios

operacionais já tinham que ser apresentados. Desta forma, foi necessário um esforço extra de modo a conseguir que estes estivessem prontos.

O restante planeamento decorreu como previsto.

Tarefa	2015				2016						
	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul
1. Solução de Inteligência Empresarial na Área da Energia	x	x	x	x	x	x	x	x	x	x	x
2. Estado da Arte do setor e soluções da empresa	x										
3. Modelo de Dados	x	x									
3.1. Análise do modelo de dados do SGC	x										
3.2. Desenho do modelo de dados		x									
4. Processo de ETL (O1)			x	x	x	x	x				
4.1. Análise do processo existente			x								
4.2. <i>Jobs</i> de extração de dados			x	x	x	x					
4.3. <i>Jobs</i> de transformação de dados I				x	x	x					
4.4. <i>Jobs</i> de transformação de dados II				x	x	x					
4.5. Otimização da performance dos <i>jobs</i>						x	x				
5. Relatórios Operacionais (O2)				x	x	x	x	x	x		
5.1. Implementação dos relatórios				x	x	x					
5.2. Validação e correção dos relatórios					x	x	x		x		
6. <i>Dashboards</i> de BI (O3)								x			
7. Escrita do relatório final									x	x	x

Figura 1.2 Planeamento do PEI

1.5 Contribuições

Uma das principais contribuições do PEI foi a concretização do objetivo O1, com o qual se garantiu um processo de ETL otimizado (tempo de execução inferior a três minutos) e em conformidade com os dados do sistema ERP que armazena os dados do SGC. Os dados extraídos não estavam coerentes com os dados existentes no SGC e o processo de ETL existente era demorado (tempo de execução superior a vinte horas).

Também a realização do objetivo O2 foi de elevada importância para o projeto SGC, uma vez que a equipa do projeto tinha uma formação agendada com o cliente, na qual se incluíam todos os relatórios operacionais requeridos pelo mesmo. Esta formação correu bem e, na generalidade, o *feedback* foi positivo.

O desenvolvimento dos *dashboards* do objetivo O3 permitiu que as provas de conceito concretizadas integrassem um conjunto diferenciador de visualizações de indicadores de desempenho relevantes para os (possíveis) clientes. Dado que o objetivo de uma prova de conceito é demonstrar as potencialidades de exploração de uma ferramenta, a inclusão destes *dashboards* trouxe uma funcionalidade extra ao sistema apresentado.

1.6 Notação adotada

O presente documento foi redigido em português, ao abrigo do novo acordo ortográfico, e todos os termos de outro idioma apresentam-se a itálico, tal como *Business Intelligence*.

1.7 Estrutura do documento

O presente relatório está dividido em quatro capítulos organizados da seguinte forma:

- No Capítulo 1 faz-se uma introdução ao tema do PEI, descreve-se a motivação para a sua realização, apresentam-se os objetivos definidos, a instituição de acolhimento, o planeamento e execução bem como as contribuições do projeto.
- No Capítulo 2 expõem-se conceitos teóricos relacionados com o tema do PEI, nomeadamente sobre BI, sistemas ERP, o processo de ETL, relatórios operacionais e *dashboards* de BI.
- No Capítulo 3 descreve-se detalhadamente o trabalho realizado, incluindo aspetos do ambiente de desenvolvimento, a concretização do processo de ETL, relatórios operacionais e *dashboards* de BI.
- No Capítulo 4 discutem-se as principais contribuições do trabalho realizado, as competências adquiridas no decorrer do estágio, bem como as principais dificuldades encontradas e trabalho futuro.

Capítulo 2

Conceitos

No presente capítulo, detalham-se conceitos predominantes do Projeto de Engenharia Informática (PEI), nomeadamente o *Business Intelligence* (BI), os sistemas *Enterprise Resource Planning* (ERP), o processo de *Extraction-Transformation-Loading* (ETL) e relatórios de apoio à decisão, nos quais se incluem os relatórios operacionais e os *dashboards*.

2.1 *Business Intelligence*

Nos últimos anos, os sistemas de informação ganharam um papel de relevo na concretização de diversas tarefas das empresas, principalmente através do registo e consulta/análise de dados.

Um dos mais importantes ativos de uma organização são os seus dados, cuja informação é utilizada normalmente com dois propósitos: a manutenção de registos operacionais, e a tomada de decisões analíticas. Os dados operacionais resultam da execução dos processos de negócio de uma organização e são armazenados em sistemas operacionais que estão otimizados para o processamento rápido e repetitivo de transações curtas (por exemplo, transações bancárias e vendas de produtos). Quanto aos dados analíticos, estes fazem parte dos sistemas de apoio à decisão (cujas transações são mais longas, complexas e personalizáveis) e devem permitir a avaliação do desempenho dos processos de negócio [6].

Numa tentativa de responder às exigências dos decisores, bem como à necessidade de implementar processos de análise de volumes de informação cada vez maiores, capazes de extrair conhecimento, surgiu o conceito de BI, os *Executive Information Systems* (EIS). Este novo conceito pretendeu colmatar as lacunas apontadas aos sistemas de apoio à decisão existentes na década de 60 (*Management Information Systems*) e de 70 (*Decision Support Systems*), nomeadamente a pouca flexibilidade dos relatórios e a pouca relevância dos dados fornecidos pelos sistemas de informação [7] [8].

Por volta de 1989, Howard Dresner definiu BI como sendo “um conjunto de conceitos e métodos para melhorar a tomada de decisões de negócio, auxiliada por sistemas de apoio à decisão” [8]. Desta definição, pode-se concluir que o BI não tem o objetivo de decidir mas de auxiliar, principalmente através de elementos visuais, a tomada de decisões por parte de decisores.

A matéria-prima das ferramentas de BI são os dados que, quando agrupados e conformados, originam informação. Da mesma forma, quando relacionada, a informação gera conhecimento. Através da análise do conhecimento extraído dos dados, é possível: i) gerar métricas para otimizar processos e métodos de trabalho; ii) estabelecer estratégias de negócio; e iii) estimar e prever oscilações de mercado [9].

Uma ferramenta de BI deve apresentar funcionalidades que permitam: i) extrair, transformar e carregar dados (processo de ETL); ii) visualizar análises de dados; iii) fazer a análise exploratória em cubos *Online Analytical Processing* (OLAP); iv) apresentar a informação em *dashboards*; v) implementar processos de *data mining*, que é o processo exploratório de grandes quantidades de dados que visa identificar padrões novos e potencialmente úteis, bem como extrair conhecimento dos mesmos [10]; e vi) criar relatórios de dados. A Figura 2.1 ilustra a arquitetura típica de uma ferramenta de BI [9].

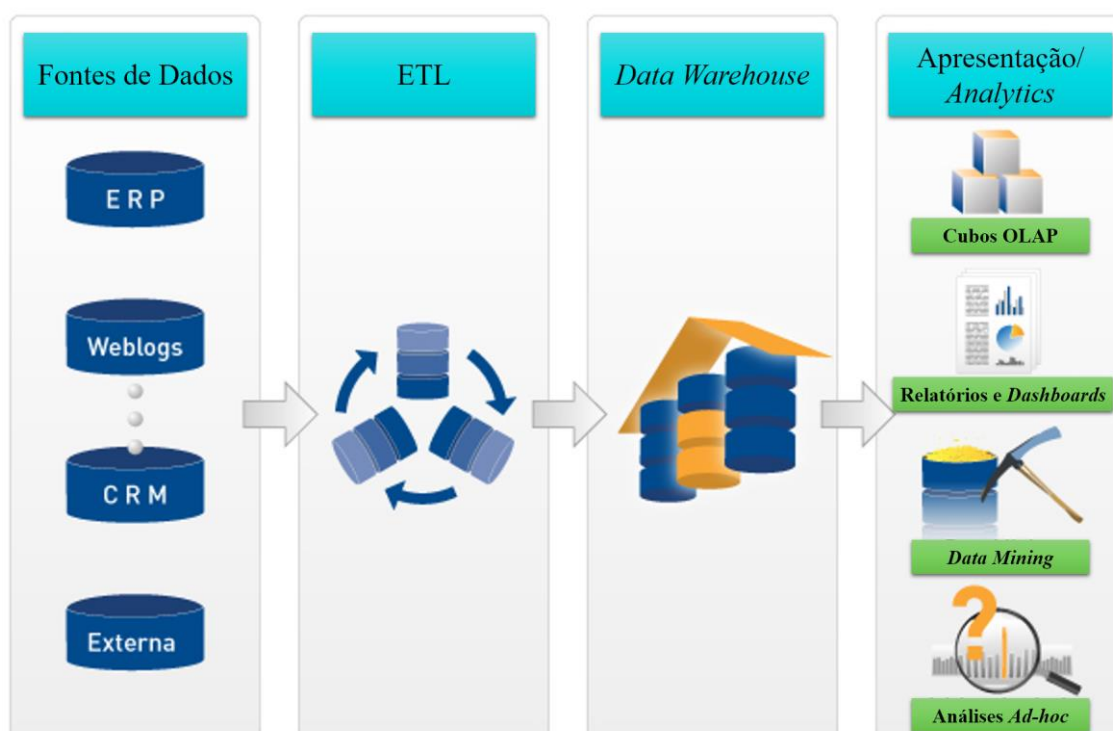


Figura 2.1 Arquitetura de uma ferramenta de BI

O *data warehouse* (DW) é um elemento chave na arquitetura de uma ferramenta de BI. Kimball e Caserta [6] apresentam a definição: “um DW é um sistema que extrai, limpa,

adapta, e fornece dados de uma (ou várias) origem num repositório de dados dimensional e, de seguida, apoia e implementa consultas e análises para efeitos de tomada de decisão”. Esta infraestrutura é um repositório central de dados que integra várias fontes de dados, criado com o objetivo de apoiar o processo de tomada de decisão. O DW pode incluir dados atuais e históricos potencialmente interessantes para os gestores (decisores) da organização, e os seus dados estão estruturados para permitir a realização de atividades de processamento analítico (*data mining*, relatórios operacionais, *dashboards*) [11] e a procura de explicações através de funcionalidades como o *drill-down* (maior grau de detalhe dos dados) e o *roll-up* (menor grau de detalhe dos dados) [9] [11].

No âmbito do DW existem ainda os conceitos de dimensão, tabela de factos e *data mart*. Uma dimensão contém atributos que descrevem os dados contidos na tabela de factos, enquanto uma tabela de factos armazena os factos sucedidos e as chaves estrangeiras para as respetivas características que se encontram nas dimensões [9] [11]. *Data mart* é um subconjunto de tabelas de dimensão que apoiam um processo de negócio, consolidando apenas informações relativas a uma área específica. A união de vários *data marts* constitui um DW único [6] [11].

No contexto do PEI, foram definidas um conjunto de dimensões e tabelas de factos que suportam os relatórios operacionais.

2.2 Processo de ETL

O processo de ETL, tal como o nome indica, representa o procedimento de Extração, Transformação e Carregamento de dados. Este processo é uma importante componente das ferramentas de BI, descrevendo a aquisição de dados provenientes de várias fontes (extração), a sua modificação para o objetivo pretendido (transformação) e a sua importação para uma base de dados ou DW, pronto a usar pelo decisor (carregamento). Os processos de ETL ocupam até 80% do esforço em projetos de BI, de modo que o seu elevado desempenho é vital para que seja capaz de processar elevadas quantidades de dados e para manter uma base de dados atualizada [12].

Extração

Nesta primeira fase, os dados são extraídos de sistemas operacionais que podem ser heterogéneos e até externos à organização. A quantidade de dados pode ser muito grande, se forem copiados todos os dados, ou reduzida, se forem excluídos quaisquer conjuntos de dados não relevantes (por exemplo, só aqueles que foram modificados desde a última extração). Por forma a não afetar negativamente o desempenho de sistemas operacionais, esta fase corre em *background* (sem interromper a execução do sistema e sem intervenção do utilizador) ou é executada em períodos de menor atividade (por exemplo, durante a noite).

Transformação

Qualquer transformação necessária ao fornecimento de dados passíveis de interpretação no âmbito do negócio é realizada neste segundo passo. Os conjuntos de dados são limpos para garantir a qualidade dos mesmos, convertidos para o esquema da base de dados destinatária (*data staging area*) e consolidados.

Relativamente a estas duas primeiras fases, importa ainda realçar que os dados permanecem numa *data staging area*, uma localização intermédia para a qual os dados das fontes de dados são copiados. Kimball e Ross [6] sugerem mesmo a colocação dos dados numa estrutura de *staging* após cada uma das principais etapas de ETL.

Carregamento

Finalmente, é necessário efetivar o carregamento dos dados para o DW. O carregamento inicial, que normalmente não é limitado por questões de tempo, distingue-se do carregamento incremental. Considerando que a primeira fase afetou os sistemas operacionais, o carregamento pode ter um enorme efeito sobre o DW. Esta questão em específico tem de ser levada em consideração no que respeita à complexa tarefa de atualizar os conjuntos de dados armazenados. Na generalidade, o carregamento incremental é uma tarefa crítica [12].

A Figura 2.2 apresenta o processo de ETL e a coordenação entre as suas fases.

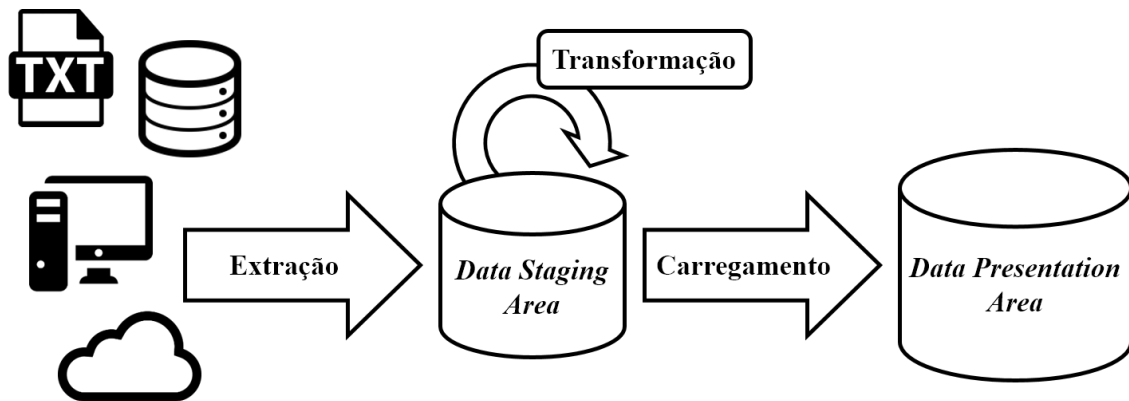


Figura 2.2 Processo de ETL

As ferramentas de ETL são peças de *software* responsáveis pela concretização das diferentes fases do processo de ETL. As principais funcionalidades destas ferramentas incluem: i) a identificação de informação relevante (por exemplo, apenas os dados novos) do lado das fontes; ii) a extração desta informação; iii) a conformação e integração da informação proveniente de múltiplas fontes num formato comum; iv) a limpeza dos dados resultantes, em termos da base de dados e de regras de negócio; e v) a propagação dos dados para o DW ou *data marts*. No entanto, por norma estes processos devem ser concluídos numa determinada janela de tempo; logo, a sua otimização é imperativa [13] [14].

Atualmente, com o crescente volume de dados, é essencial melhorar as capacidades de processamento de dados dos processos de ETL sob pena de estes demorarem horas ou dias a terminar, podendo levar a tomadas de decisões de negócio imprecisas. Com um fluxo de dados de ETL otimizado, as alterações aos dados nas fontes podem refletir-se mais rapidamente no DW e os utilizadores podem efetivar decisões de negócio mais exatas utilizando os dados mais recentes [14] [15].

2.3 Sistemas ERP

Os sistemas ERP, também denominados sistemas integrados de gestão, são sistemas de *software* orientados ao negócio que permitem que uma empresa: i) automatize e integre a maioria dos seus processos de negócio; ii) produza e aceda à informação em tempo-real; e iii) partilhe dados e práticas comuns a todos os seus departamentos [16] [17]. Um pacote de aplicações de *software* de ERP é um conjunto de módulos pré-construídos e integrados, que dão resposta a todas as funções de negócio e que têm flexibilidade suficiente para configurar e personalizar as funcionalidades fornecidas pelo pacote para atender às necessidades específicas da empresa [18].

Uma das principais características de um sistema ERP é a integração da informação, que é inserida uma única vez num dos seus módulos e que fica disponível em todo o sistema imediatamente, estando organizada de forma a poder ser utilizada em tempo-real pelos vários elementos decisores da organização, melhorando a qualidade global da informação disponibilizada.

As suas funcionalidades vão para além de um *software* concebido para cada departamento da empresa. Este tipo de sistema é desenvolvido partindo da perspetiva da organização como um todo. Toda a informação é guardada uma única vez no sistema, em vez de o ser em vários sistemas diferentes, ficando a informação registada numa única base de dados, contínua e coerente. Os vários departamentos não necessitam de estabelecer ligações a diferentes sistemas para obterem respostas, pois toda a organização utiliza os mesmos dados, ficando a informação imediatamente disponível ao utilizador [16] [19].

Para além das vantagens em termos de integração, os sistemas ERP permitem aumentar os benefícios intrínsecos de cada um dos seus módulos, uma vez que cada um integra as “*best practices*” de cada uma das funções de negócio. As informações são armazenadas e processadas em cada módulo do ERP, os quais representam um conjunto de funções que servirá um ou vários departamentos da empresa [19].

No entanto, os sistemas ERP não são criados para cada organização em específico: as empresas que desenvolvem o sistema (por exemplo, a SAP SE¹) fazem-no de forma generalizada e a utilização do sistema requer que cada organização faça os devidos ajustes para que os seus modelos de processos sejam incorporados. O sucesso dos sistemas ERP baseia-se no princípio da reutilização das suas funcionalidades. Por exemplo, a SAP desenvolveu o SAP R/3 com base nas semelhanças mais essenciais observadas no funcionamento das empresas pertencentes a uma indústria, e a principal implementação que os utilizadores do SAP têm de realizar é a seleção dos processos desta biblioteca que são necessários à sua empresa [18]. As principais áreas de aplicação dos pacotes de *software* de ERP são Finanças, Logística e Recursos Humanos [18] [19].

No contexto do PEI, a fonte de dados do processo de ETL desenvolvido foi um ERP, nomeadamente o *SAP Industry Solution for Utilities* (SAP IS-U).

2.4 Relatórios de apoio à decisão

Os relatórios de apoio à decisão facilitam a tomada de decisão no âmbito das organizações, fornecendo um conjunto de indicadores que permitem avaliar o desempenho das suas diversas áreas e motivar a concretização de modificações, caso sejam necessárias [9] [11] [20].

No âmbito do presente PEI, foram concretizados dois tipos de relatórios de apoio à decisão: relatórios operacionais e *dashboards* de BI.

Relatórios Operacionais

Os relatórios operacionais são um dos tipos de relatórios mais comuns no apoio à decisão. Contêm grandes quantidades de dados operacionais, que podem estar armazenados num ERP, organizados numa estrutura muito bem definida, que são essenciais para as operações de produção. Por norma, estes dados estão num nível de detalhe muito elevado, ou seja, no nível transacional, e são concretizados com o objetivo de suportar as atividades diárias de uma empresa [20].

Estes relatórios devem fornecer: i) informação atual ao utilizador; ii) informação detalhada, cujos dados estão no menor nível de granularidade²; e iii) flexibilidade para permitir que os utilizadores finais possam criar as suas visualizações específicas dos dados [21].

¹ <http://go.sap.com/index.html>

² Granularidade representa o nível de detalhe dos dados.

Dashboards de BI

Os *dashboards* são apresentações visuais da informação mais importante necessária para alcançar um ou vários objetivos de negócio, consolidada e ajustada a um único ecrã para permitir o acompanhamento rápido do negócio da empresa [22]. Os *dashboards* integram a informação de múltiplas áreas de negócio e apresentam gráficos e dados calculados que mostram o desempenho atual comparado com as métricas pretendidas.

Estes elementos fornecem uma perspetiva visual de medidas de desempenho corporativa, tendências e exceções. As referidas medidas são também designadas por *Key Performance Indicators* (KPIs). Os KPIs são medidas quantificáveis que permitem perceber se determinados objetivos estão a ser atingidos, facilitando o processo de tomada de decisão [11].

2.5 Sumário

Neste capítulo, apresentaram-se os principais conceitos no âmbito do PEI. Inicialmente, explicitou-se o conceito de BI, tanto numa vertente histórica da sua evolução como tecnológica, detalhando também os conceitos de *data warehouse*, tabela de dimensão, tabela de factos e *data mart*. Seguidamente, descreveu-se o conceito de sistema ERP que, no contexto do presente PEI, assume o papel de fonte de dados, apresentando algumas das suas características diferenciadoras face a outros sistemas. Em terceiro lugar, detalhou-se o conceito de processo de ETL, onde se especificaram as suas fases e principais funcionalidades. Por fim, incluíram-se os relatórios de apoio à decisão, detalhando aqueles que fizeram parte do PEI, nomeadamente os relatórios operacionais e os *dashboards*.

No capítulo seguinte, descreve-se o trabalho desenvolvido, incluindo o modelo de dados concretizado, o processo de ETL e respetiva otimização, bem como a implementação dos relatórios operacionais e a construção de *dashboards*.

Capítulo 3

Trabalho realizado

Neste capítulo, descreve-se o trabalho realizado durante o estágio. Apresenta-se o ambiente de trabalho, bem como o modelo de dados criado, o processo de *Extraction-Transformation-Loading* (ETL) desenvolvido e os relatórios operacionais e *dashboards* concretizados.

3.1 Ambiente de trabalho

Nesta secção descrevem-se as ferramentas utilizadas ao longo do Projeto de Engenharia Informática (PEI), a organização do trabalho da equipa do Sistema de Gestão Comercial (SGC) da Novabase em que se inseriu o trabalho e o processo de desenvolvimento de *software*.

3.1.1 Ferramentas utilizadas

Faz parte da política da empresa dar prioridade à utilização de *software open source* face a *software* proprietário. Assim, durante o estágio, as ferramentas utilizadas foram maioritariamente *open source*. Ao iniciar o estágio, as ferramentas de desenvolvimento dos relatórios já tinham sido seleccionadas por elementos hierarquicamente superiores da equipa. Detalham-se abaixo as referidas ferramentas:

SQL Power Architect 1.0.³

Este *software* de modelação de dados foi utilizado no desenho do modelo relacional (objetivo O1), tendo facilitado bastante o referido processo uma vez que já existia uma versão primária da base de dados que serve os relatórios operacionais, permitindo a importação das tabelas e respetivos campos.

Na Figura 3.1 apresenta-se a interface desta ferramenta, na qual se pode constatar a presença de alguns elementos de relevo, nomeadamente: 1) a listagem de tabelas da

³ <http://www.sqlpower.ca/page/architect>

base de dados relacional; 2) a área de trabalho em que o modelo relacional foi desenvolvido; e 3) um conjunto de botões de acesso rápido (por exemplo, *zoom-in/zoom-out*, adição de nova tabela, adição de nova relação).

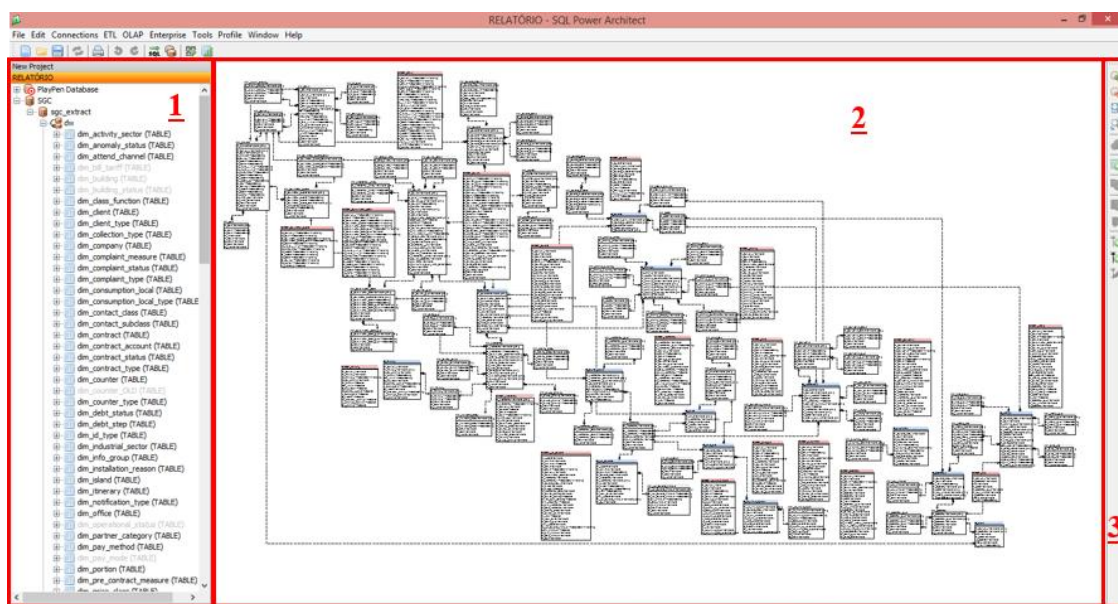


Figura 3.1 Interface de utilização do *SQL Power Architect*

PostgreSQL 9.4.4⁴

É o SGBD que suporta as bases de dados implementadas (objetivo O1). O PostgreSQL é uma das mais avançadas ferramentas *open source* do género, disponibilizando recursos como *triggers*, índices, *views*, e suportando os tipos de dados mais comuns (no PEI, utilizaram-se os tipos *character varying*, *number*, *decimal*, *date* e *boolean*) [23].

A utilização da linguagem procedimental *Procedural Language/PostgreSQL* (PL/pgSQL) foi também um fator de interesse, uma vez que esta linguagem foi alvo de estudo e aplicação numa cadeira anterior do plano de estudos do mestrado a que o presente documento diz respeito. Através desta linguagem, foi possível concretizar um conjunto de funções para controlar a execução dos processos de ETL e para facilitar o cálculo de alguns campos dos relatórios operacionais desenvolvidos.

Talend Open Studio for Data Integration 5.6.2⁵

O Talend foi a ferramenta selecionada para implementar os processos de ETL e integração de dados. Este *software* permite a concretização de extrações e tratamento de

⁴ <https://www.postgresql.org/>

⁵ <https://www.talend.com/products/talend-open-studio>

dados, a implementação de *jobs*⁶ que facilitam a execução dos processos em horários menos sobrecarregados, bem como o carregamento dos dados para a respetiva base de dados.

O Talend é uma ferramenta baseada no Eclipse⁷ e em Java⁸, de fácil utilização principalmente devido ao seu ambiente de desenvolvimento gráfico e funcionamento através de *drag-and-drop*, permitindo que a orquestração de processos seja concretizada de forma intuitiva e rápida. Cada *job* desenvolvido representa uma classe de Java, cujo código é gerado automaticamente pelo Talend à medida que é construído.

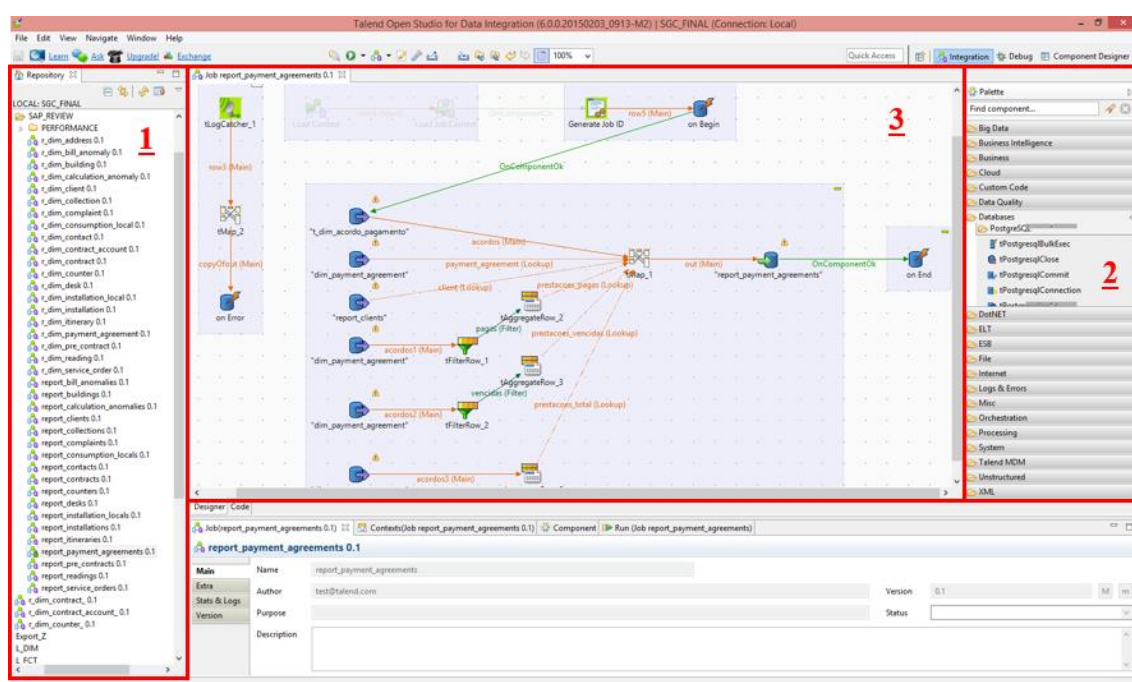


Figura 3.2 Interface de utilização do Talend

Na Figura 3.2 apresenta-se a interface de utilização desta ferramenta, destacando as áreas mais relevantes.

1. Repositório: apresenta a listagem dos *jobs* desenvolvidos;
2. *Palette*: disponibiliza os diferentes componentes técnicos passíveis de serem utilizados num *job* (por exemplo, *tPostgresqlInput* e *tMap*);
3. Área de trabalho: zona em que se constrói o *job*, fazendo uso dos componentes disponibilizados pela *Palette*, bem como de um conjunto de conectores que despoletam uma determinada ação (em detalhe na Subsecção 3.3.3).

⁶ *Job* é o conjunto de componentes interligados por conexões que representa uma sequência de acontecimentos entre os mesmos.

⁷ <https://eclipse.org/>

⁸ <https://www.java.com/en/>

SAP IS-U⁹

O conjunto de processos que fazem parte do SGC foram desenvolvidos no sistema SAP IS-U. O SAP IS-U é um sistema de vendas e informação que apoia todos os processos de negócio e serviços utilitários de uma empresa dos serviços públicos [24].

A aplicação *standard* do SAP (SAP R/3) assenta sobre a premissa que, dentro das diferentes indústrias, existe um amplo conjunto de funcionalidades comuns a todas, podendo ser agrupadas em três módulos: Finanças, Logística e Recursos Humanos. No entanto, também é verdade que praticamente todas as indústrias possuem requisitos próprios que são específicos das empresas que atuam nessa indústria.

Assim, por forma a suportar as especificidades das diferentes indústrias, o SAP fornece um conjunto de soluções que endereçam os requisitos únicos das mesmas e são complementares ao SAP base. Um dos componentes de SAP específicos para uma indústria é o referido SAP IS-U.

O SAP tem uma funcionalidade abrangente, mas, no que toca a empresas individuais, o verdadeiro desafio está em poder personalizá-lo rapidamente de acordo com os seus requisitos específicos. Assim sendo, o SAP fornece as ferramentas para que uma empresa possa adaptar o sistema aos seus requisitos específicos, configurando os parâmetros em tempo de implementação. Este procedimento designa-se parametrização [18].

No contexto do processo de ETL, este sistema teve o papel de fonte de dados, uma vez que os dados que compõem os relatórios provêm do mesmo.

Pentaho Report Designer 5.2.0.0¹⁰

A interface dos relatórios operacionais (objetivo O2) foi concretizada através do Pentaho. Esta ferramenta *open source* permite criar relatórios detalhados com base em dados preparados, que podem provir de diversas fontes. A sua interface gráfica de utilizador é de fácil utilização, através de *drag-and-drop*, e permite criar relatórios com a precisão de um *pixel*.

Este *software*, em conjunto com um servidor fornecido com o mesmo, possibilita a publicação de relatórios *online*, o que facilitou a sua integração com o *Unified Front-End* (UFE). Para além disso, o Pentaho permite a exportação dos dados dos relatórios para diversos formatos, onde se inclui o Excel. Esta funcionalidade é relevante pois um dos requisitos definidos foi a exportação dos dados dos relatórios para Excel.

⁹ <http://go.sap.com/solution/industry/utilities.html>

¹⁰ <http://community.pentaho.com/projects/reporting/>

Na Figura 3.3 encontra-se a interface de utilização do Pentaho, da qual fazem parte:

1. *Palette*: disponibiliza todos os elementos gráficos passíveis de serem adicionados a um relatório (por exemplo, imagens, campos de texto, totalizadores);
2. Área de trabalho: zona que permite criar e pré-visualizar os relatórios;
3. Ferramenta de Estrutura: apresenta todos os elementos visuais e não visuais presentes nos relatórios;
4. Ferramenta de Propriedades: lista todas as propriedades da seleção ativa, permitindo a modificação das propriedades comuns a diversos elementos seleccionados de uma só vez (por exemplo, tipo de letra, alinhamento do texto, cor de preenchimento, negrito, itálico, sublinhado).

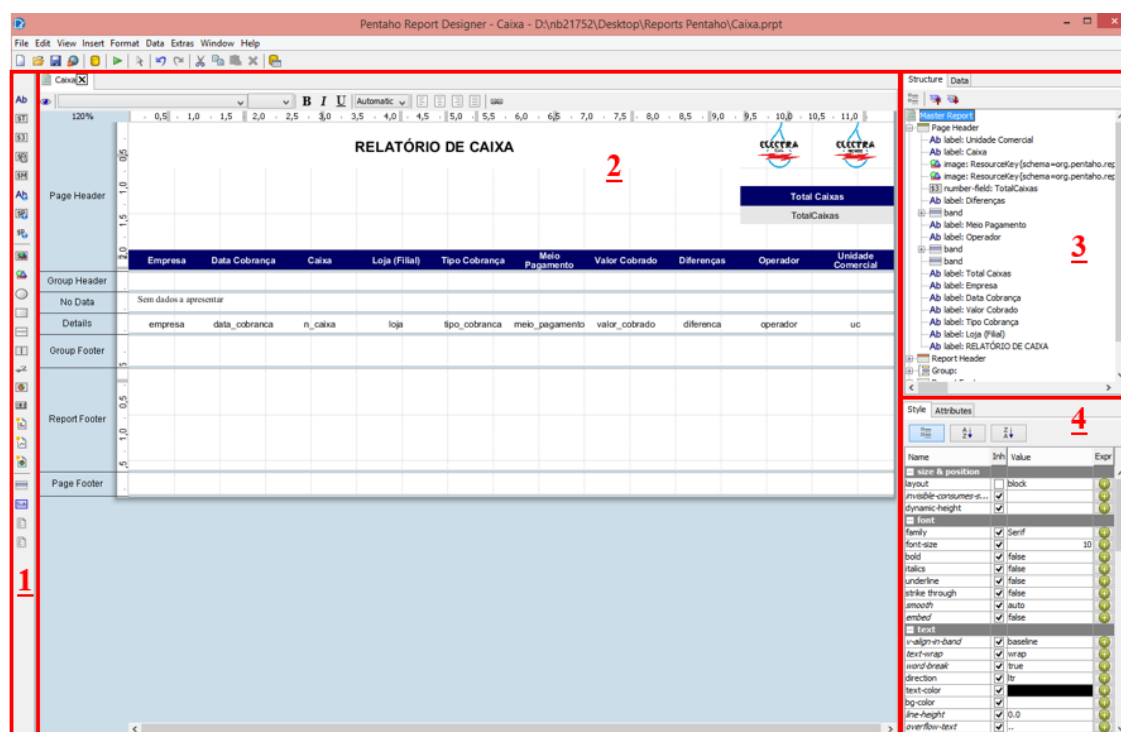


Figura 3.3 Interface de utilização do Pentaho

Oracle Business Intelligence Enterprise Edition 11g (OBIEE)¹¹

Os *dashboards* criados para as provas de conceito dos clientes da Zâmbia e do Gana foram realizados no OBIEE por requisito de ambas as empresas. Esta ferramenta existe sob a forma de uma aplicação Web e permite a criação de *dashboards*, gráficos e tabelas, com relativa facilidade.

¹¹ <http://www.oracle.com/us/solutions/business-analytics/business-intelligence/enterprise-edition/overview/index.html>

O funcionamento do OBIEE incide sobre um servidor aplicativo (Oracle BI Server) e um repositório (Oracle BI Repository), este último dividido em três camadas (apresentação, modelo de negócio e física). O servidor não armazena dados, guarda apenas metadados das fontes de dados que utiliza na geração e execução de interrogações *Structured Query Language* (SQL). O *BI Repository* contém todos os metadados das camadas de negócio de um repositório.

Apresenta-se na Figura 3.4 a interface de edição de um *dashboard*.

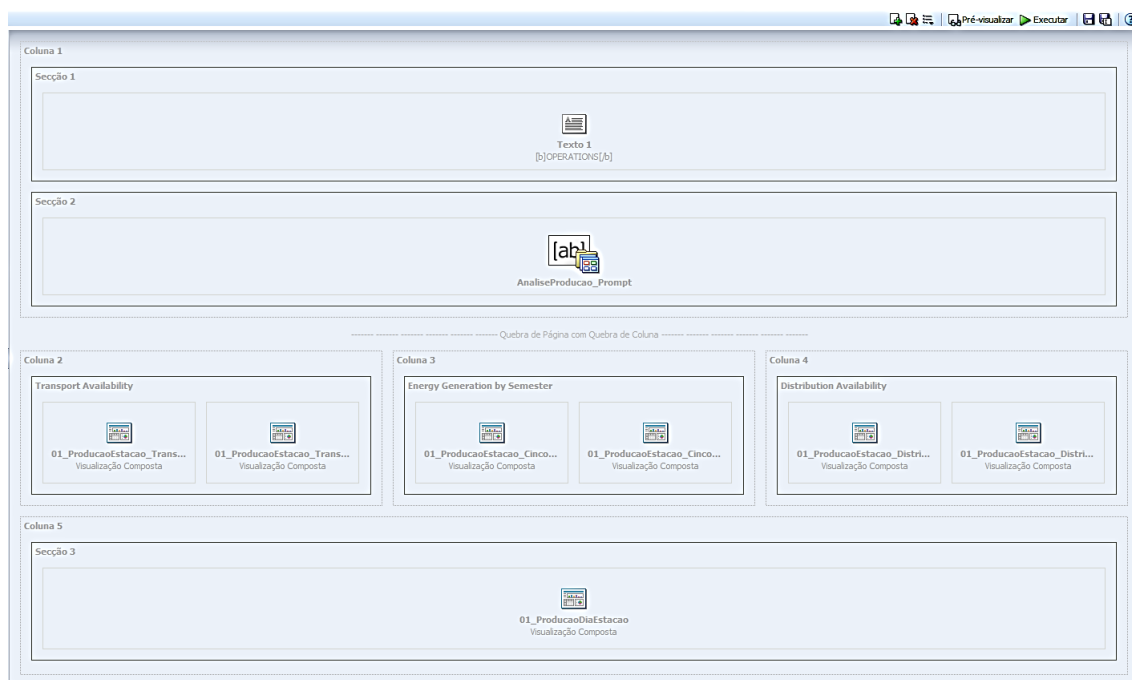


Figura 3.4 Interface de utilização do OBIEE (edição de *dashboard*)

3.1.2 Organização do trabalho na equipa

O SGC é composto por uma equipa de trabalho multidisciplinar, cujos elementos estão organizados consoante a sua área de intervenção, nomeadamente a funcional e a técnica.

Os consultores funcionais são especialistas em processos de negócio, e responsáveis por desenhar a solução de implementação de SAP, concretizar parametrizações, avaliar requisitos, e especificar e testar processos. Cada funcional está dedicado a um ou vários módulos do SAP IS-U. Os elementos da equipa técnica são especialistas em tecnologias de informação, responsáveis pela implementação das especificações funcionais e dos processos em SAP, desenvolvimento da interface de utilização (UFE) bem como a componente de relatórios. No Apêndice I apresenta-se um organograma da equipa de trabalho do SGC.

O trabalho desenvolveu-se em ambiente informal, com toda a equipa a trabalhar em mesas contíguas, permitindo que decisões relativas ao desenvolvimento do projeto fossem concretizadas pessoalmente, ou via correio eletrónico caso algum elemento estivesse deslocado. Esta interação facilita uma maior dinâmica na troca de ideias e discussão de opções entre a equipa, agilizando o processo de desenvolvimento.

Para além destas formas de comunicação, são também realizados dois tipos de reuniões:

- **Pontos de Situação** que decorrem entre o gestor do projeto e as equipas funcional e técnica, cujos objetivos são reportar o estado de desenvolvimento em que o projeto se encontra e discutir os próximos acontecimentos.
- **Reuniões Periódicas** que acontecem entre o cliente e a equipa funcional, com os objetivos de apresentar as implementações e progressos realizados pelas equipas, reformular requisitos e discutir pontos críticos dos processos de negócio do projeto.

A Figura 3.5 ilustra o funcionamento da equipa de desenvolvimento.

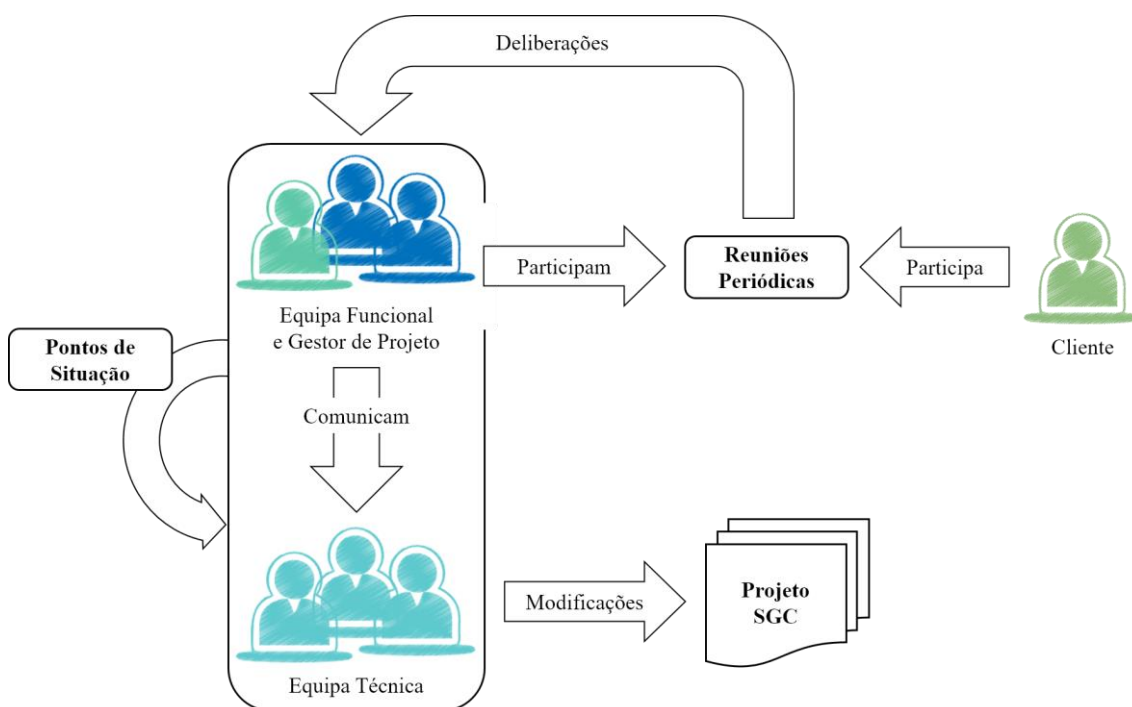


Figura 3.5 Esquema de funcionamento da equipa

3.1.3 Processo de desenvolvimento de *software*

O desenvolvimento de *software* foi orientado segundo o Scrum, uma Metodologia Ágil que segue uma abordagem iterativa e incremental para o desenvolvimento de *software* e gestão de projetos. Esta metodologia enquadrou-se bem com as características do projeto, cujos prazos de entrega foram curtos e os requisitos sofreram algumas alterações em qualquer momento.

A estrutura do Scrum considera a existência de três papéis, nomeadamente:

- **Product Owner**, que representa os interesses dos *stakeholders* e é o responsável pela definição de requisitos do produto, por atribuir prioridades do *product backlog*, por garantir o valor do trabalho realizado pela equipa de desenvolvimento e por aceitar ou recusar o resultado de cada *sprint*¹². Este papel foi desempenhado pelo cliente do SGC.
- **Scrum Master**, responsável por assegurar a concretização dos objetivos/requisitos definidos no planeamento, por facilitar a colaboração entre funções e áreas da equipa e por garantir que o processo de Scrum está a ser seguido. Este papel pertenceu ao gestor de projeto, Sérgio Pinto.
- **Equipa de desenvolvimento**, grupo multifuncional de pessoas responsável, pelo desenvolvimento e entrega das funcionalidades do produto. Os elementos das equipas funcional e técnica do SGC representaram este papel.

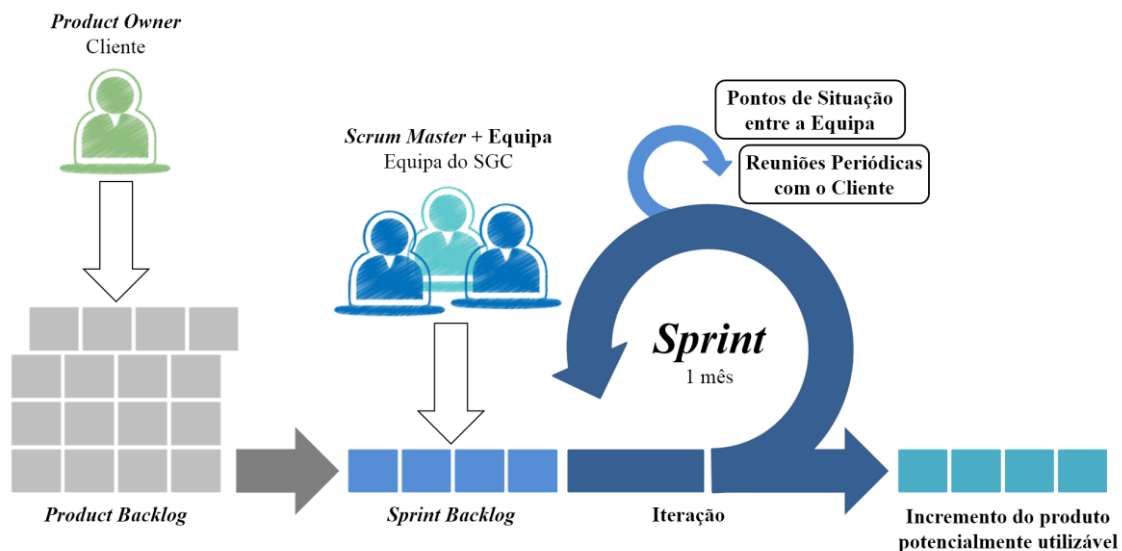


Figura 3.6 Fluxo do processo de desenvolvimento de *software* Scrum

Deste processo de desenvolvimento de *software* fazem ainda parte dois tipos de artefactos:

- **Product Backlog**, uma lista de requisitos/funcionalidades solicitadas pelo cliente, cujos itens do *backlog* estão ordenados pela sua prioridade. Pode ser alterada pelo *product owner* a qualquer momento, desde que as alterações não afetem o período do *sprint* em curso.

¹² *Sprint* representa uma iteração do projeto, um período definido em que deve ser implementado um ou mais requisitos definidos no *backlog*. Tipicamente, dura entre duas a quatro semanas, sendo que na equipa do SGC a duração foi de um mês [26].

- ***Sprint Backlog***, uma sublista do *product backlog*, resultante do *sprint planning meeting*, que contém os requisitos/funcionalidades que se pretende concretizar no *sprint* seguinte, nomeadamente os mais prioritários.

Cada *sprint* é precedido de uma reunião de planeamento (*sprint planning meeting*) entre o *product owner* e a equipa de desenvolvimento, para definir quais os itens do *product backlog* que serão implementados no *sprint* seguinte (tal como ilustrado na Figura 3.5). Os itens selecionados passam a pertencer ao *sprint backlog*. Durante cada *sprint*, a equipa de desenvolvimento implementa as tarefas definidas na reunião e atualiza um ficheiro com as modificações executadas até à data do fim do *sprint* [25] [26].

Tal como se pode verificar nas figuras 3.5 e 3.6, a organização do trabalho da equipa esteve bem alinhada com o processo de desenvolvimento de *software*.

3.2 Caracterização dos dados dos processos

Uma vez que existem elementos comuns aos vários objetivos do trabalho realizado, começa-se por apresentar uma caracterização dos dados dos processos do SGC.

Por forma a obter-se uma visão ampla e completa dos dados e das suas relações, bem como do contexto em que estes se encontram inseridos, desenvolveu-se um modelo relacional com base nos campos definidos para os relatórios operacionais, no qual se detalham as relações entre dimensões e possíveis tabelas de factos. A realização deste modelo permitiu também reduzir o número de campos que seriam extraídos e que podiam ser obtidos através de outras tabelas. Por último, o modelo serve ainda como documentação, muito útil na passagem de conhecimento e na concretização de projetos idênticos no futuro.

Os relatórios operacionais solicitados foram:

- Relatório de Anomalias de Cálculo;
- Relatório de Anomalias de Faturação;
- Relatório de Caixas;
- Relatório de Clientes;
- Relatório de Cobranças;
- Relatório de Contactos;
- Relatório de Contadores;
- Relatório de Contratação;
- Relatório de Cortes por Dívida;
- Relatório de Dívidas;
- Relatório de Faturação;

- Relatório de Instalações;
- Relatório de Itinerários;
- Relatório de Leituras;
- Relatório de Locais de Consumo;
- Relatório de Locais de Instalação de Equipamentos;
- Relatório de Ordens de Serviço;
- Relatório de Planos de Pagamento;
- Relatório de Pré-Contratação;
- Relatório de Prédios;
- Relatório de Reclamações.

Dada a quantidade de entidades existentes no SGC e o número de relatórios (cerca de 21), este modelo ficou relativamente grande. Assim, optou-se por apresentar primeiro um modelo de dados simplificado, que fornece uma representação global da base de dados, ao qual se segue um conjunto de partições do mesmo para permitir a sua pormenorização. No sentido de diminuir a redundância, as entidades que fazem parte de vários diagramas são detalhadas uma única vez e os nomes dos atributos utilizam um conjunto de prefixos que permite a sua fácil identificação, nomeadamente:

- ID: identificador principal da entidade;
- DT: data;
- CD: código (identificador utilizado no SAP IS-U);
- DS: descritor/descrição;
- FL: *flag*, indicador de verdadeiro/falso.

Importa ainda realçar a omissão nos diagramas seguintes de dois campos de controlo que existem em todas as dimensões, designadamente *fl_active*, uma *flag* booleana que define se o registo ainda se encontra ativo/válido; e *dt_update*, que define a última data em que o registo foi atualizado.

Na Figura 3.7 encontra-se o modelo de dados simplificado do SGC, que expõe os conceitos base do sistema, e que está dividido através de um conjunto de cores para identificar os diagramas mais pequenos que se descrevem nas subsecções seguintes. Estas divisões definiram-se com base nos processos de negócio do SGC, em que a verde se encontra a Contratação e Serviço ao Cliente, a cor-de-rosa a Gestão de Equipamentos, a roxo a Gestão de Leituras, a azul a Faturação e Cobranças e a amarelo a Gestão de Trabalho.

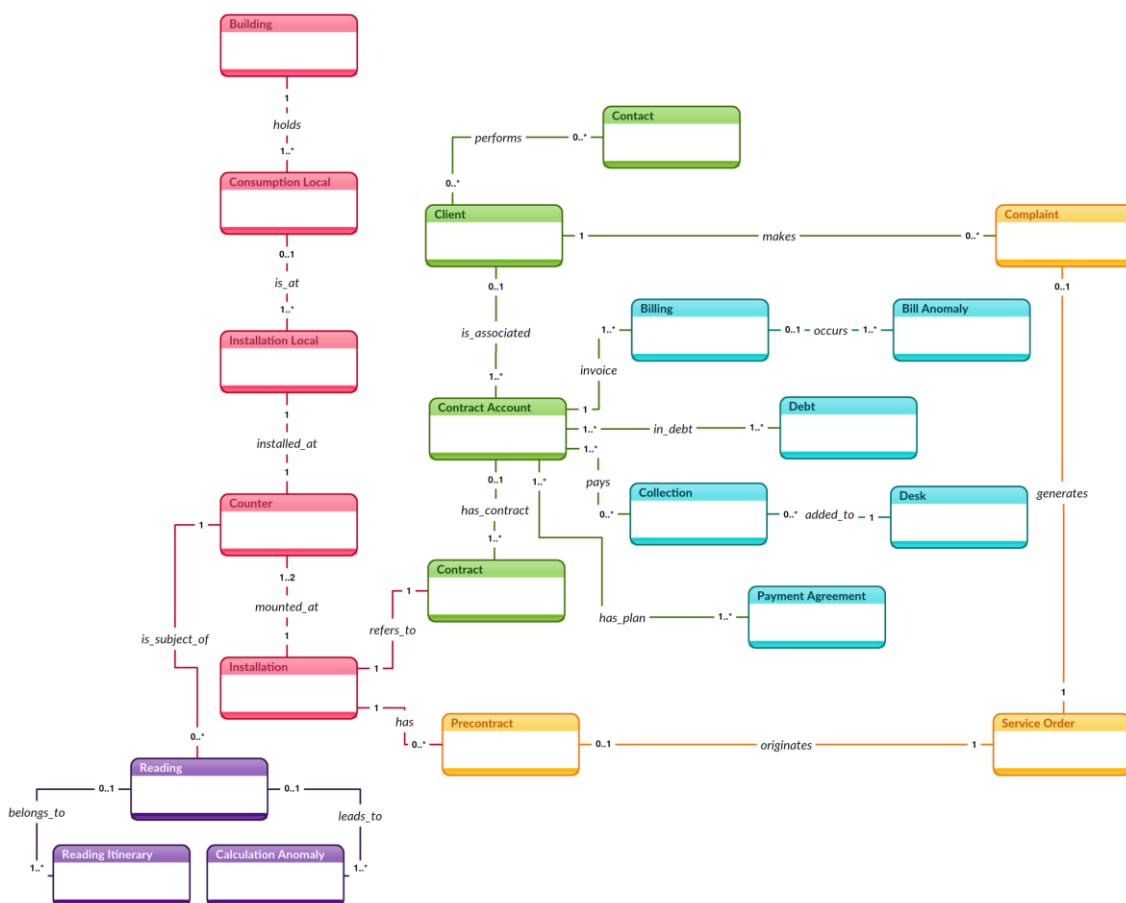


Figura 3.7 Modelo de dados simplificado do SGC

De realçar também que os diagramas que se seguem apresentam as dimensões presentes na Figura 3.7 destacadas a verde para facilitar a sua contextualização face ao modelo simplificado.

3.2.1 Contratação e serviço ao cliente

O processo de negócio do âmbito da Contratação e Serviço ao Cliente tem como objetivo a definição de condições entre o cliente e a empresa, nomeadamente através da realização de um contrato de prestação de serviços.

Na Figura 3.8 encontra-se o modelo de dados detalhado relativo a este processo, no qual se detalham as relações entre as entidades *Client*, *Contract Account*, *Contract* e *Contact*. Um **cliente** está associado a uma **conta-contrato**, que por sua vez pode ter no máximo dois **contratos** (um de água e um de eletricidade). Os clientes efetuam **contactos** de diversos tipos (*contact class* e *contact subclass*), através de meios de atendimento (*attend channel*), que são analisados por um utilizador do SGC (*user*).

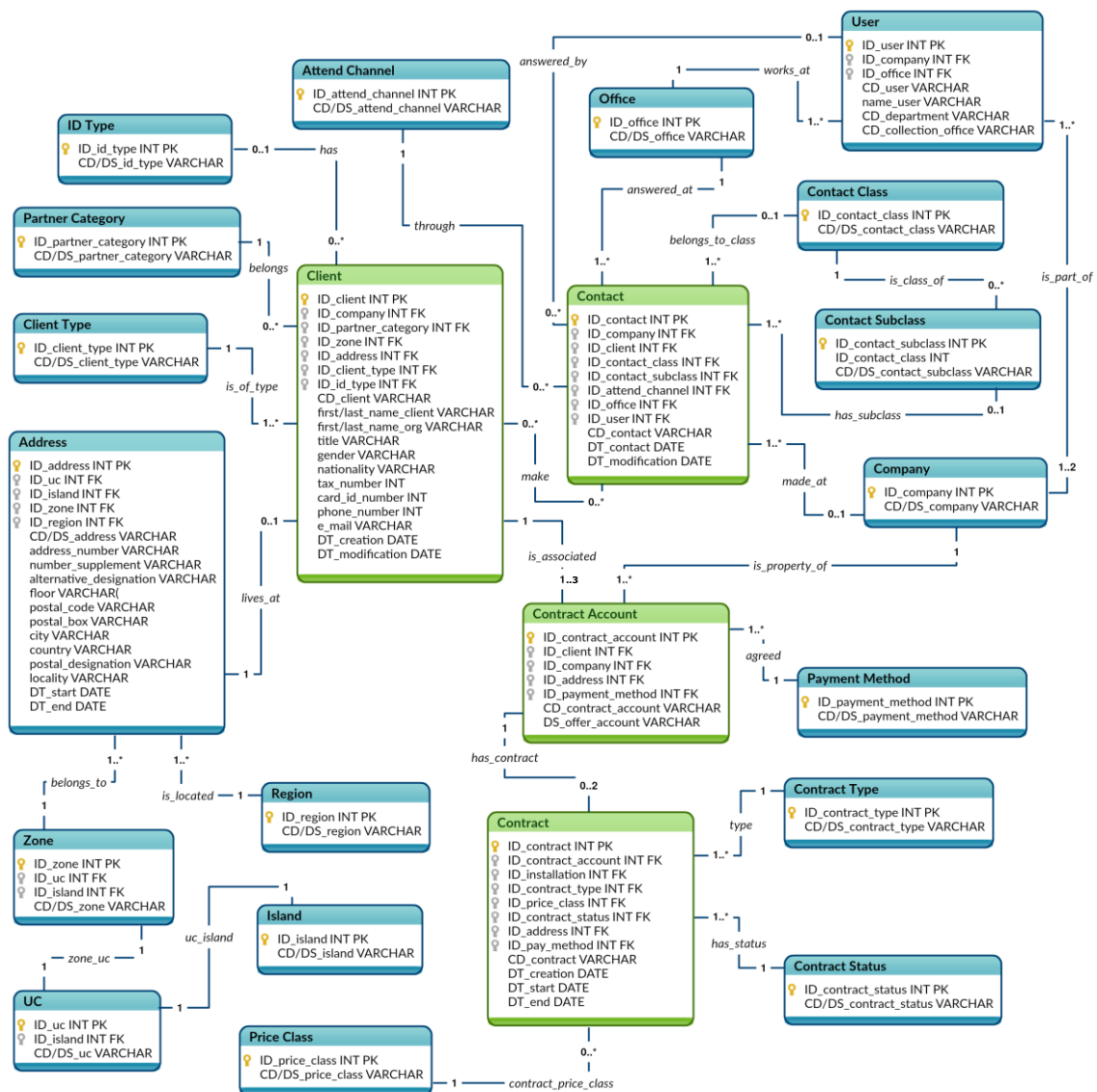


Figura 3.8 Modelo de dados detalhado relativo ao processo de negócio da Contratação e Serviço ao Cliente

3.2.2 Gestão de equipamentos

O processo de negócio relativo à Gestão de Equipamentos pretende garantir a manutenção das infraestruturas e a operacionalidade das mesmas. No modelo detalhado da Figura 3.9 destacam-se as entidades *Building*, *Consumption Local*, *Installation Local*, *Installation* e *Counter*. Um **prédio** representa um agregado de vários **locais de consumo** (como um prédio comum), que podem ser vistos como os diversos apartamentos que constituem esse edifício. Para além disto, um **local de consumo** pode ter vários **locais de instalação** (locais de colocação de um contador), onde se instalam os **contadores** (no máximo dois, um por cada setor de atividade – *activity sector*). Uma **instalação** relaciona um contador e um local de consumo.

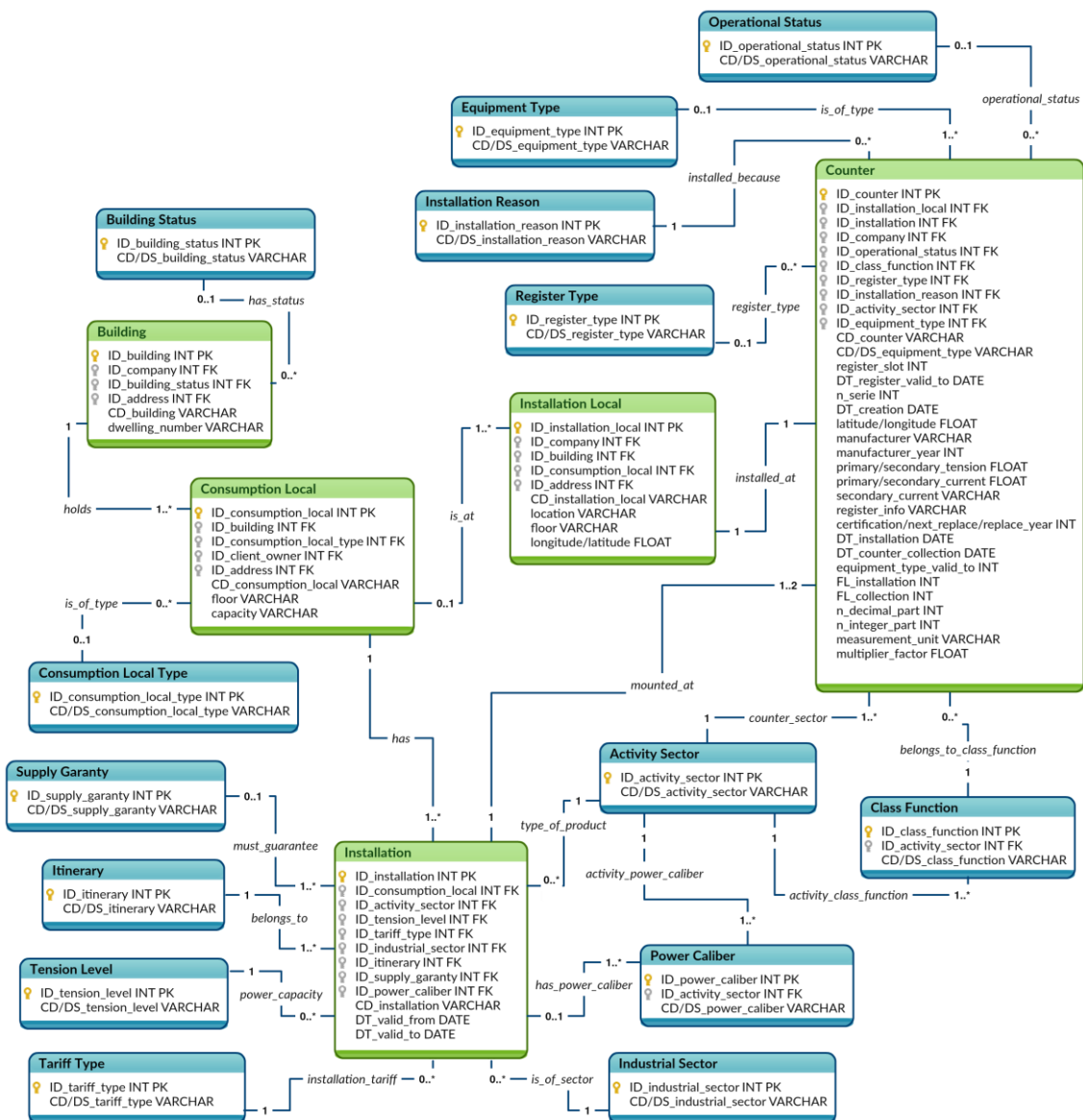


Figura 3.9 Modelo de dados detalhado relativo ao processo de negócio da Gestão de Equipamentos

3.2.3 Gestão de leituras

O processo de negócio do âmbito da Gestão de Leituras tem como objetivo garantir o planeamento e operacionalidade das leituras realizadas aos equipamentos, bem como as equipas que procedem a tal. Na Figura 3.10 destacam-se as entidades *Reading*, *Calculation Anomaly* e *Reading Itinerary*. Uma **leitura** é o registo dos consumos de um cliente, contabilizados num contador. Esta leitura está associada a um **itinerário de leitura**, composto por um conjunto de contratos (*portion*) e por um itinerário terrestre (*itinerary*). Se a leitura não for corretamente feita, origina-se uma **anomalia de cálculo**, que depois é alvo de correção e, caso seja necessário, procede-se a ajuste na faturação.

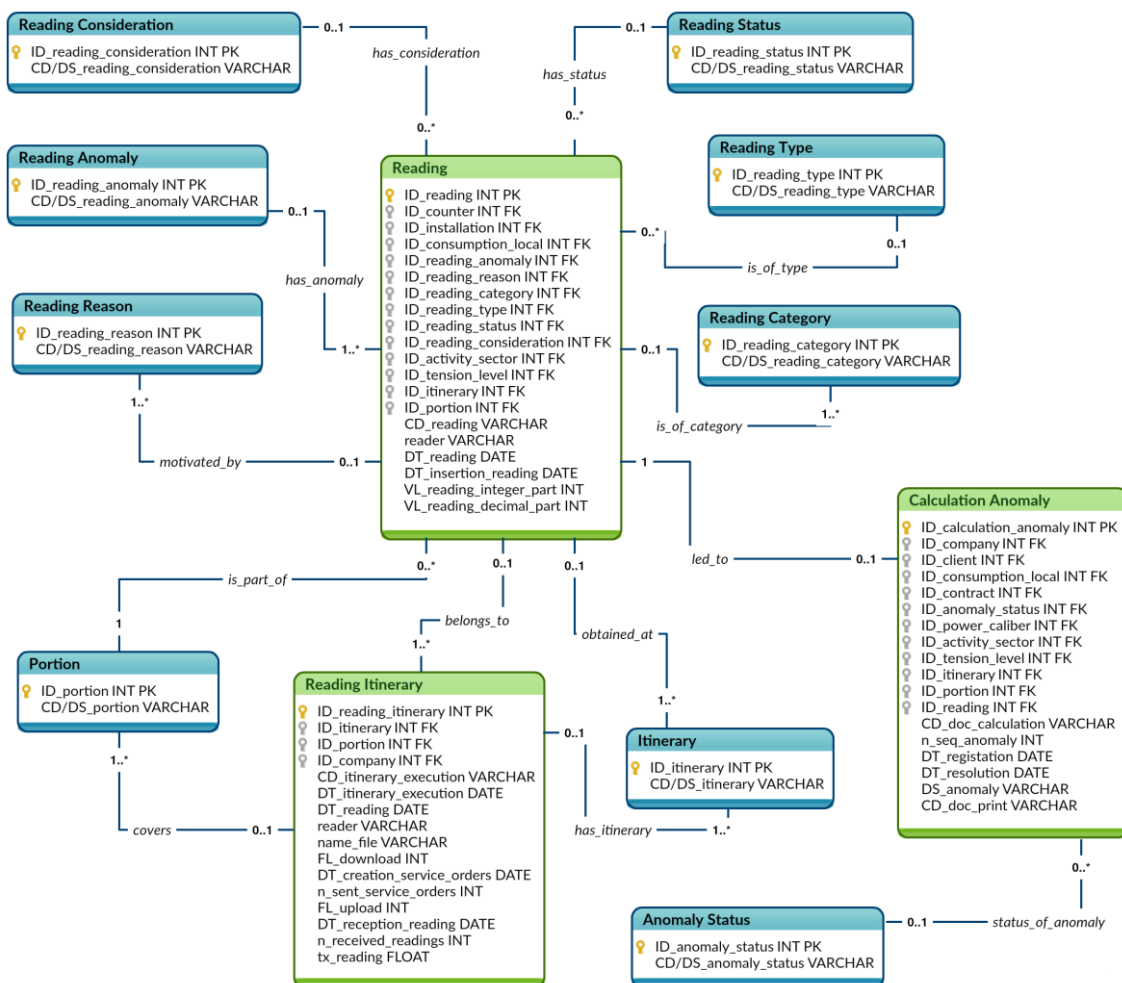


Figura 3.10 Modelo de dados detalhado relativo ao processo de negócio da Gestão de Leituras

3.2.4 Faturação e cobranças

Quanto ao processo de negócio de Faturação e Cobranças, este tem como objetivos: i) no caso da Faturação, o cálculo dos consumos, a sua faturação, impressão e entrega da fatura ao cliente; ii) no caso das Cobranças, asseverar que os pagamentos referentes à prestação de serviços por parte da empresa decorrem sem prejuízo para ambas as partes.

Na Figura 3.11 destacam-se as entidades *Payment Agreement*, *Collection*, *Collection Document*, *Debt*, *Billing*, *Bill Anomaly*, *Billing Document*, *Desk*. Cada conta-contrato tem um **plano de pagamento** associado, que define a periodicidade de pagamento dos consumos realizados. Quando o cliente efetua um pagamento, que é registado numa **caixa**, gera-se uma **cobrança**, à qual está associado um **documento de cobrança**, e quando o pagamento está em falta cria-se um processo de **dívida**. Podem ainda ocorrer **anomalias de faturação** (por exemplo, quando uma leitura é malfeita e o cliente paga consumos que não fez). A **faturação** representa a relação entre uma cobrança e uma fatura.

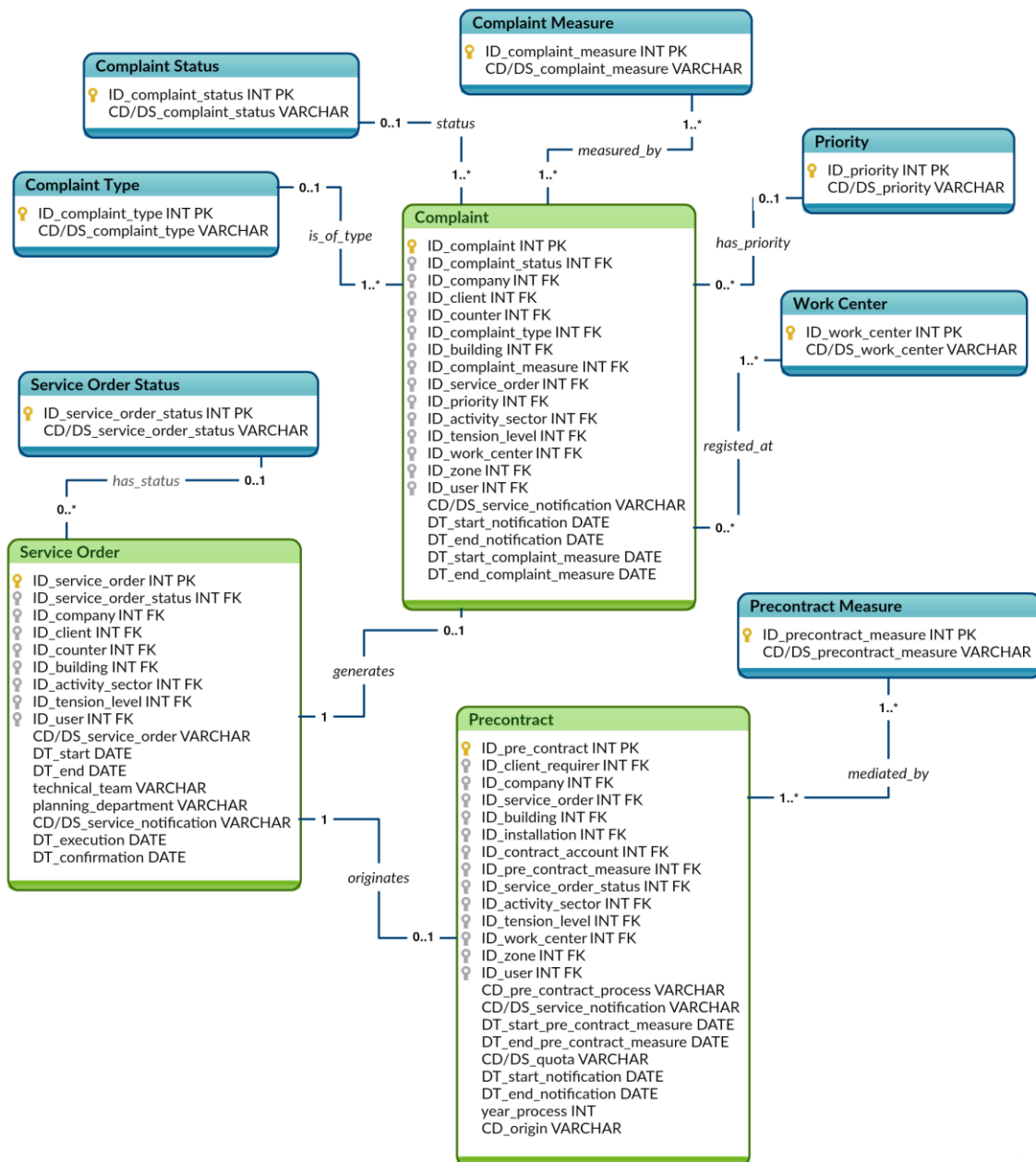


Figura 3.12 Modelo de dados detalhado relativo ao processo de negócio da Gestão de Trabalho

Após o desenvolvimento do modelo de dados descrito nas subsecções acima, iniciou-se o desenvolvimento do processo de ETL.

3.3 Desenvolvimento do processo de ETL

Esta secção diz respeito ao objetivo O1, em que se pretendia o desenvolvimento de um processo de ETL de dados armazenados num sistema ERP, eficiente, ágil e que resulte em dados de qualidade e em conformidade com o modelo de dados do SGC.

O processo de ETL desenvolvido para o SGC visa extrair os dados necessários do SAP IS-U; transformá-los, uniformizando o conteúdo dos campos de texto, eliminando espaços desnecessários, tornando o conteúdo de alguns campos inteligível,

incorporando identificadores unívocos sequenciais em todas as tabelas, relacionando códigos identificadores de SAP entre tabelas, mapeando identificadores sequenciais entre tabelas, agregando tabelas; e carregar os dados tratados para novas tabelas e *views* para serem utilizados pelos decisores. Na Figura 3.13 apresenta-se o fluxo de dados entre as ferramentas utilizadas (descritas na Subsecção 3.1.1) na concretização do processo de ETL.

Figura 3.13 Fluxo de dados do processo de ETL

Relativamente aos dados do SGC, importa ainda distinguir dados-mestre, dados transacionais e domínios. Considera-se que:

- ii) Dados transacionais são aqueles que resultam dos processos de negócio (por exemplo, faturas, leituras de contadores, contactos) [5];
- iii) Domínios definem intervalos de valores [27]. No contexto dos relatórios, os domínios apresentam (quase exclusivamente) informação textual descritiva (por exemplo, tipo de cliente, tipo de cobrança).

3.3.1 Preparação do ambiente de desenvolvimento

Antes de começar o desenvolvimento, foi necessário preparar um conjunto de elementos indispensáveis ao processo de ETL, nomeadamente as *data staging areas*, o contexto no Talend e a parametrização no SAP (descrita na secção 3.3.2).

A base de dados que serve os relatórios operacionais do SGC foi estruturada de maneira a definir uma separação entre os dados extraídos e os dados transformados. Para tal, concretizaram-se dois *schemas*¹³ que permitem a organização das tabelas, *views* e funções consoante a etapa de ETL (extração e transformação) em que os dados foram obtidos. Para além destes *schemas*, criou-se também uma tabela de controlo de *jobs*, que regista diversas informações sobre a execução dos *jobs* (no total, 185 *jobs*) do processo de ETL, que são importantes para a recuperação de erros.

Relativamente ao Talend, procedeu-se à definição do que se designa por contexto. Um contexto é um conjunto de variáveis cujo valor pode ser alterado em tempo de execução e que permite que os *jobs* sejam executados com diferentes parâmetros, de acordo com o ambiente em que estão a ser executados. A definição do contexto é bastante útil aquando da existência de diferentes ambientes de execução (por exemplo, de Produção e de Qualidade), em que existem diferentes bases de dados e, consequentemente, diferentes parâmetros de conexão às mesmas. Em alternativa, poder-se-iam criar duas cópias do *job*, cada qual configurada com diferentes parâmetros de conexão, mas esta abordagem duplicaria o código e tornaria a manutenção dos *jobs* mais difícil.

O contexto foi definido como metadados do Repositório do Talend e contém informação relativa às *data staging areas* e ao SAP IS-U. Na Figura 3.14 apresenta-se o menu de definição do contexto, no qual se podem observar as variáveis definidas. Cada variável tem um valor associado, de acordo com o ambiente de execução a que se refere. Na *drop-down list* “Default context environment” encontram-se os diferentes ambientes de execução definidos no contexto (excetuando o *Default*). Estes diferentes

¹³ No contexto PostgreSQL, *schema* representa uma coleção de objetos de bases de dados, isto é, tabelas, *views*, índices, funções, sequências, tipos de dados. Uma base de dados pode ter vários *schemas*.

ambientes de execução permitem que os *jobs* possam ser executados em máquinas distintas, com diferentes configurações.

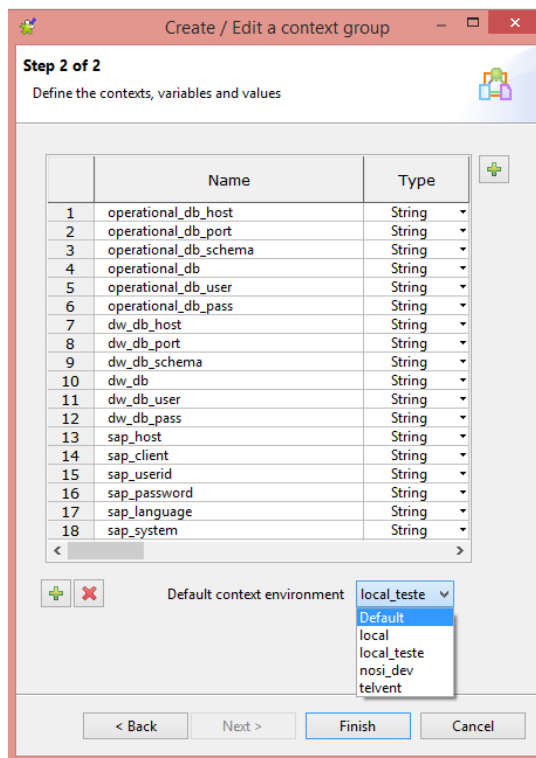


Figura 3.14 Menu de definição do contexto do Talend

3.3.2 Parametrização de campos SAP

Após alguma discussão com a equipa técnica de SAP relativamente ao impacto da extração de dados para os relatórios, optou-se pela distribuição da carga entre os intervenientes, nomeadamente entre o ERP e o Talend. Inicialmente pensou-se que o SAP extrairia a informação de tabelas completas para ficheiros de texto, cujos dados seriam posteriormente relacionados e tratados pelo Talend. No entanto, esta abordagem, para além de requerer maior processamento do lado do ERP, também requeria que fossem implementados os processos de extração dos dados por um técnico de SAP, o que, por questões de tempo e esforço, não era possível.

Assim, optou-se por uma solução alternativa: os técnicos de SAP desenvolveram uma função *Remote Function Call* (RFC) em *Advanced Business Application Programming* (ABAP, a linguagem de programação utilizada para desenvolver programas no SAP) que lê um conjunto de tabelas de parametrização e cria extratores dinâmicos de dados. Os dados extraídos são depois devolvidos pela função RFC a um componente específico de SAP utilizado nos *jobs* do Talend.

No entanto, esta alternativa requer o preenchimento das respetivas tabelas de parametrização. Nestas tabelas foram definidos: i) os campos que contêm os dados a

serem extraídos; ii) as tabelas onde os dados se encontram, bem como os critérios de seleção e *arrays* para exclusão ou inclusão; iii) os valores para os *arrays* de exclusão ou inclusão de dados. Esta lógica é aplicável tanto a domínios como a dados-mestre e transacionais.

3.3.3 Extração de dados

O processo de ETL foi desenvolvido no Talend, sendo composto por um conjunto de *jobs*. Um *job* ou *subjob* é composto por um ou vários componentes logicamente ligados entre si através de conexões. Os componentes são os elementos funcionais dos *jobs*, responsáveis pela execução de uma determinada operação (por exemplo, inserção de linhas na base de dados, ordenação dos dados, filtro dos dados). Estes elementos estão disponíveis na *Palette* do Talend, agrupados segundo o seu âmbito de funcionamento (por exemplo, componentes de Oracle, de MySQL, de PostgreSQL). Os componentes utilizados foram:



tExtractDelimitedFields: gera múltiplas colunas a partir de uma única coluna com campos delimitados.



tFileInputDelimited: lê um ficheiro, linha-a-linha, com os campos separados por um delimitador.



tFileOutputDelimited: escreve para um ficheiro, linha-a-linha, com os campos separados por um delimitador.



tFileOutputRaw: escreve dados para um ficheiro.



tJava: permite adicionar código Java personalizado de modo a integrá-lo no *job* do Talend.



tMap: permite realizar *joins*, filtragem de colunas ou linhas, transformações e concretizar múltiplos *outputs*.



tPostgresqlBulkExec: carrega os dados eficientemente para um ficheiro.



tPostgresqlInput: lê uma tabela do PostgreSQL e extrai os campos com base numa interrogação SQL.



tPostgresqlOutput: insere ou atualiza linhas numa tabela do PostgreSQL.



tPostgresqlOutputBulk: escreve um ficheiro temporário para o carregamento em massa no PostgreSQL.



tPostgresqlRow: executa uma *query* numa base de dados PostgreSQL.



tSAPConnection: cria uma conexão ao sistema SAP.



tSAPInput: extrai dados do sistema SAP.



tSortRow: ordena um fluxo de dados, permitindo a ordenação de múltiplas colunas, ordenação ascendente/descendente e alfabética.

Para além dos componentes, os *jobs* são também compostos por conexões. Estas conexões podem ser de diferentes tipos, de acordo com os dados a processar, o tipo de *output* ou a sequência lógica do *job*. Existem dois tipos de conexões principais: i) *Row*, que lida com dados, e ii) *Trigger*, que define a sequência de processamento (não há dados tratados). Estes dois tipos de conexões podem ainda ser de diferentes subtipos, de entre os quais se detalham aqueles que foram utilizados na implementação dos diversos processos. Relativamente a conexões *Row*, empregaram-se os subtipos: a) *Main*, que passa o fluxo de dados de um componente para outro, percorrendo cada linha e lendo os dados de *input* tendo em conta o *schema*¹⁴ definido nas propriedades do componente; e b) *Lookup*, utilizado apenas em caso de múltiplos fluxos de *input* (como o tMap), que permite relacionar um componente de fluxo secundário com um componente de fluxo principal. Quanto a conexões *Trigger*, utilizaram-se os subtipos: c) *On Subjob Ok*, que despoleta o *subjob* seguinte exclusivamente se o anterior tiver terminado sem erros; e d) *On Component Ok*, que ativa o componente seguinte apenas se a execução do componente anterior tiver sido concluída sem erros.

Os primeiros *jobs* a ser desenvolvidos foram os responsáveis pela extração dos dados dos domínios, uma vez que estes se encontravam estáveis e, portanto, dificilmente sofreriam alterações nos seus campos.

Estes *jobs* de extração (cerca de 57, um para cada domínio) foram todos concretizados com base na mesma lógica (descrita abaixo), seguindo um fluxo de operações. Para além da definição deste fluxo cada componente implicou ainda a definição de um conjunto de parâmetros específicos. O *job* desenvolvido pode constatar-se na Figura 3.15, neste caso referente a Tipos de Cliente, e traduz-se na seguinte sequência de ações:

1. Gera-se um número aleatório, que é depois utilizado como identificador unívoco na tabela de controlo de *jobs*.
2. Adiciona-se uma nova linha de registo à tabela de controlo de *jobs*.
3. Abre-se uma conexão ao SAP IS-U do SGC, com base nas configurações definidas no componente e no contexto.
4. Extraem-se os dados, de acordo com um conjunto de parâmetros definidos no componente. Estes parâmetros, nomeadamente o nome do domínio e os respetivos campos, referem-se aos valores de *input* da função RFC do SAP do SGC.

¹⁴ No contexto Talend, *schema* define o conjunto de campos de *input* e *output* dos componentes. Qualquer componente que suporte o fluxo de dados tem um *schema* associado. [28]

5. Uma vez que os dados provenientes do SAP vêm numa *string* única, com os campos separados por um caracter delimitador, este componente faz a separação dos campos de acordo com um *schema* definido.
6. Inserem-se as linhas de dados na correspondente tabela da *data staging area*.
7. Atualiza-se a linha de registo do *job* na tabela de controlo de *jobs*, sinalizando o seu término a uma determinada hora.

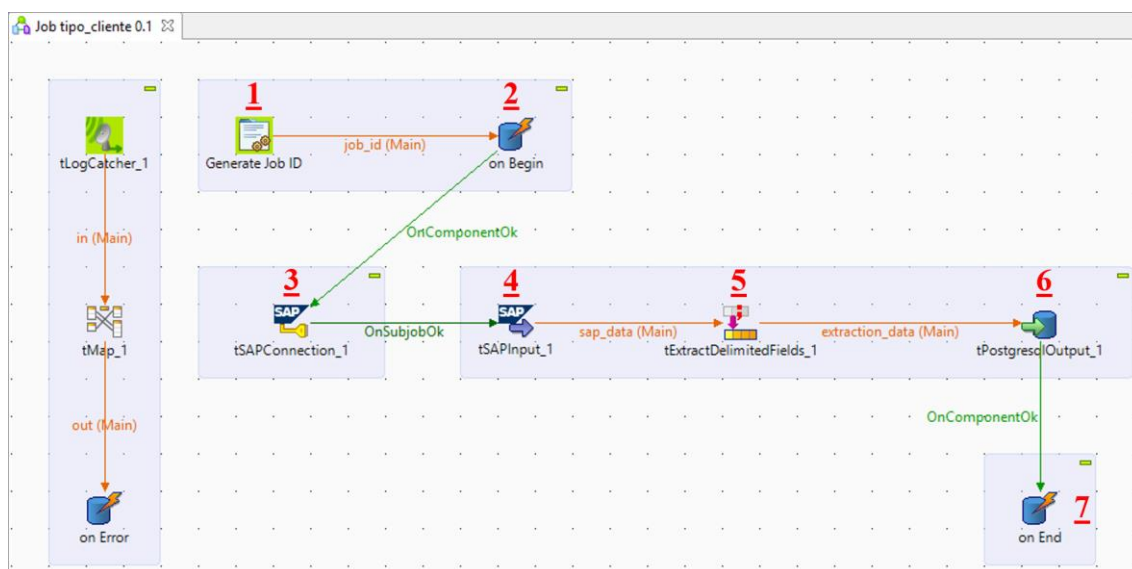


Figura 3.15 Job de extração de dados de um domínio

Os *jobs* de extração de dados-mestre e transacionais (cerca de 28), que servem os relatórios operacionais em causa, são em tudo equivalentes aos dos domínios.

3.3.4 Transformação de dados

Após a extração dos dados, procedeu-se à sua transformação. Algumas transformações foram concretizadas através de *views* sobre as respetivas tabelas de extração, permitindo eliminar alguma carga de processamento do Talend. As referidas transformações visaram a formatação de datas, a eliminação de espaços em branco (recorrente na extração de campos de SAP), a alteração de tipos de dados, a capitalização de palavras (principalmente em domínios, para efeitos de apresentação nos filtros do Pentaho), a ordenação alfabética das descrições que são apresentadas nos filtros (de forma a que os utilizadores identifiquem mais facilmente a opção pretendida) e a melhoria de inteligibilidade de alguns campos (por exemplo, alteração dos valores F/M para Feminino/Masculino).

Nesta etapa encontraram-se alguns desafios, nomeadamente quanto à formatação da data, uma vez que os campos de data extraídos vêm num formato diferente do pretendido (ano/mês/dia ao invés de dia/mês/ano); também com os campos decimais foi

necessário perceber qual o tipo de dados correspondente na base de dados, de forma a não se perderem casas decimais com arredondamentos.

Mais uma vez, os *jobs* de transformação de domínios (cerca de 59) foram os primeiros a ser desenvolvidos, neste caso devido a serem necessários aos *jobs* dos dados-mestre e transacionais. O *job* desenvolvido está ilustrado na Figura 3.16 e segue o seguinte fluxo de operações:

1. Gera-se um número aleatório, que é depois utilizado como identificador unívoco na tabela de controlo de *jobs*.
2. Adiciona-se uma nova linha de registo à tabela de controlo de *jobs*.
3. Leem-se todas as linhas da tabela do domínio (neste caso, a *view* de extração) e passam-se os respetivos campos para o componente seguinte.
4. Mapeiam-se os campos. Relativamente aos domínios, salvo dois ou três casos (um deles ilustrado na Figura 3.17), esta correspondência é direta pois a tabela resultante não inclui dados de outras tabelas.
5. Ordena-se a coluna de descrição alfabeticamente (colocando este passo do lado do Talend, não é necessário ordenar no Pentaho, permitindo um menor tempo de processamento dos relatórios).
6. Inserem-se as linhas de dados na correspondente tabela da *data staging area*. Ao realizar estas inserções, é também adicionado um identificador único a cada linha, através de uma sequência numérica concretizada ao nível de cada tabela.
7. Atualiza-se a linha de registo do *job* na tabela de controlo de *jobs*, sinalizando o seu término a uma determinada hora.

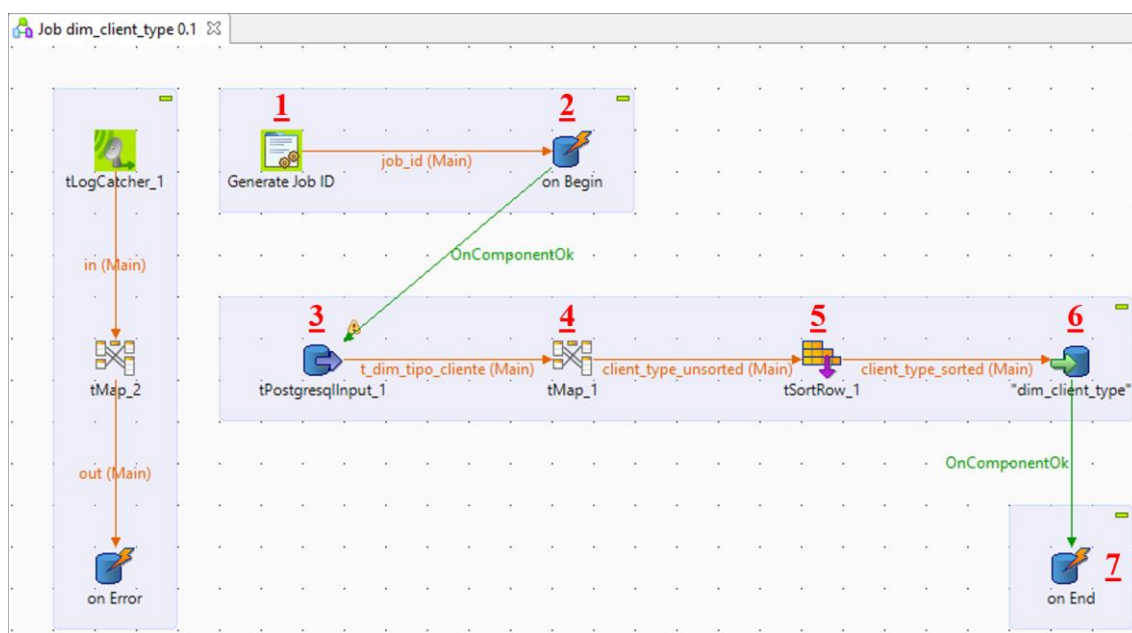


Figura 3.16 *Job* de transformação de dados do domínio Tipo de Cliente (ver Figura 3.8)

Tal como referido, houve algumas exceções face ao fluxo apresentado, uma das quais se ilustra na Figura 3.17. Uma vez que cada Subclasse de Contacto tem uma Classe de Contacto associada (ver Figura 3.8), optou-se por concretizar a transformação de ambos os domínios num único *job*. Desta forma, substituíram-se na dimensão Subclasse os códigos do sistema que identificam a sua Classe, pelo identificador sequencial único correspondente, num único *job*, garantindo que este mapeamento ocorre sempre sequencialmente.

Para o domínio Classe de Contacto, o fluxo de operações decorre como descrito (acima da Figura 3.16); no entanto, após o término do *subjob* correspondente, inicia-se um novo *subjob* em que o *input* do mapeamento (tMap) são duas tabelas: *t_dim_sub_classe_contato*, com os dados extraídos do domínio Subclasse, e *dim_contact_class*, com os dados transformados no *subjob* anterior. Este mapeamento (com múltiplas tabelas de *input*) equipara-se a um LEFT OUTER JOIN entre as duas tabelas numa base de dados.

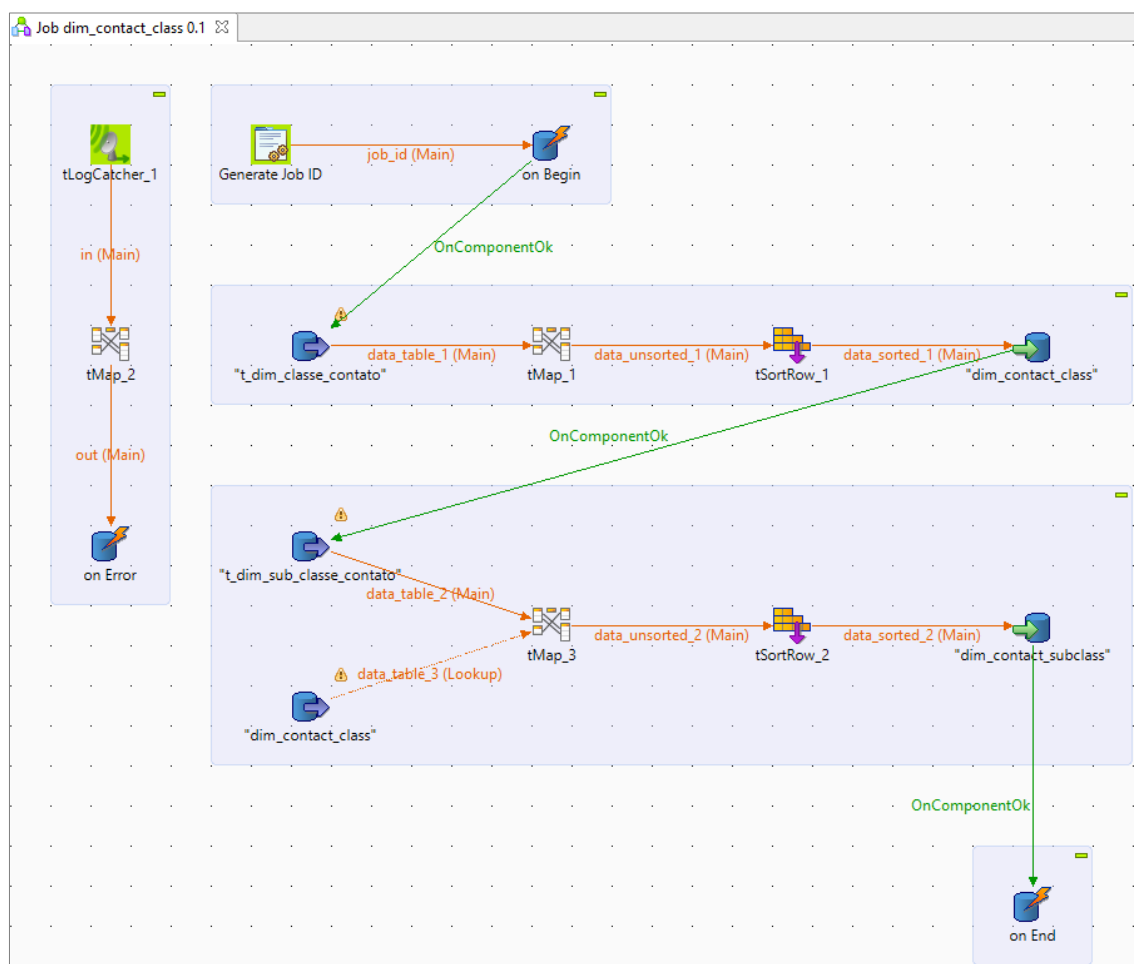


Figura 3.17 Job de transformação de múltiplos domínios (Classe de Contacto e Subclasse de Contacto)

Relativamente aos *jobs* de transformação dos dados-mestre e transacionais (cerca de 22), o processo foi idêntico, embora a ordenação não tenha sido realizada e o

mapeamento dos campos tenha sido mais complexo, dada a grande quantidade de tabelas de *input* (a título de exemplo, nos casos das Reclamações e Pré-Contratos chegam a ser dezasseis tabelas). Uma vez que o principal objetivo desta fase I de transformação (ver Figura 3.13) foi substituir todos os códigos de sistema pelos identificadores únicos correspondentes dos domínios (em casos mais complexos, de dados-mestre/transacionais), o mapeamento dos campos foi também bastante linear. Esta fase foi importante pois permitiu concretizar relações entre dados e garantir a conformidade dos dados dos relatórios.

A Figura 3.18 exhibe o mapeamento dos campos sobre clientes entre as múltiplas tabelas de *input* do SGC e a tabela de *output* na *data staging area*. As ligações roxas que estão junto à margem esquerda representam a relação entre os campos da tabela principal (no topo) e as restantes tabelas incluídas no mapeamento, sendo equivalente à definição da cláusula ON quando se realiza um LEFT OUTER JOIN entre tabelas na linguagem SQL. As linhas amarelas no centro da figura mapeiam os campos do lado esquerdo com os do lado direito, definindo que informação de entrada fará parte da tabela de saída.

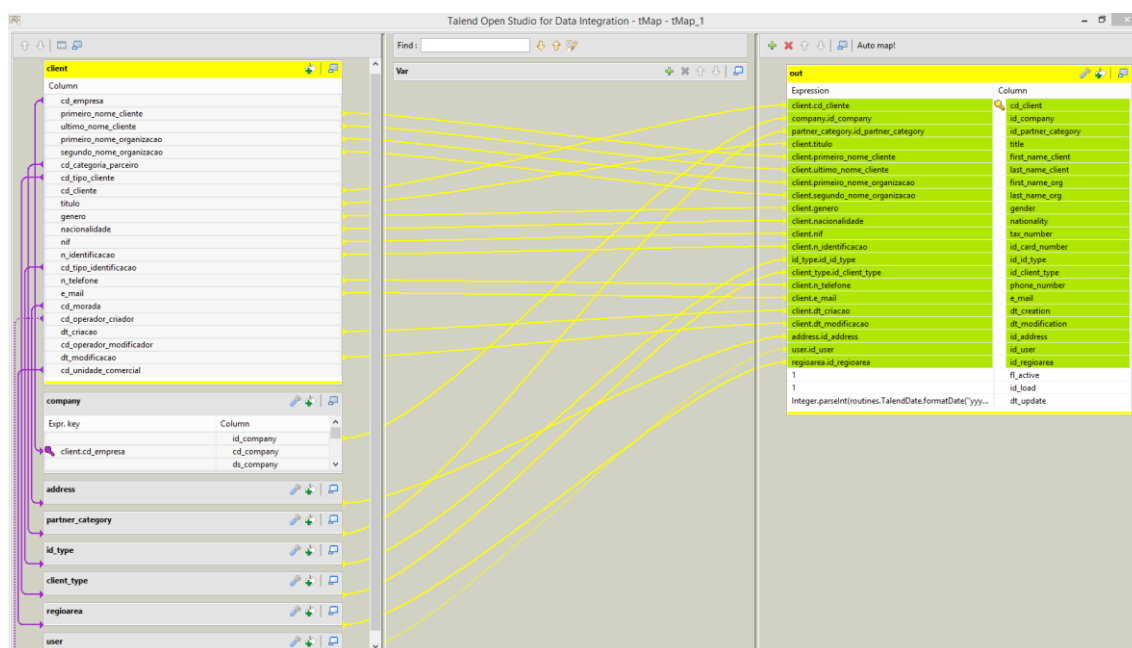


Figura 3.18 Mapeamento de campos na fase I de transformação

De seguida, procedeu-se ao desenvolvimento dos *jobs* da fase II de transformação (ver Figura 3.13). Uma vez que cada relatório contempla informação localizada em várias tabelas de domínios, dados-mestre e dados transacionais, o principal objetivo desta fase foi agregar dados que estão em diferentes tabelas numa única tabela. Deste modo, o Pentaho pode aceder à respetiva tabela do relatório sem necessitar de processar uma *query* com múltiplos JOIN (para relacionar os dados) ao carregar um relatório operacional. Um exemplo é o relatório de Clientes (listado na Secção 3.2), que reúne a

informação dos dados-mestre Cliente e Morada e de quatro domínios (ver Figura 3.8). O fluxo de operações é equivalente ao da fase I de transformação, diferenciando-se na complexidade dos mapeamentos realizados.

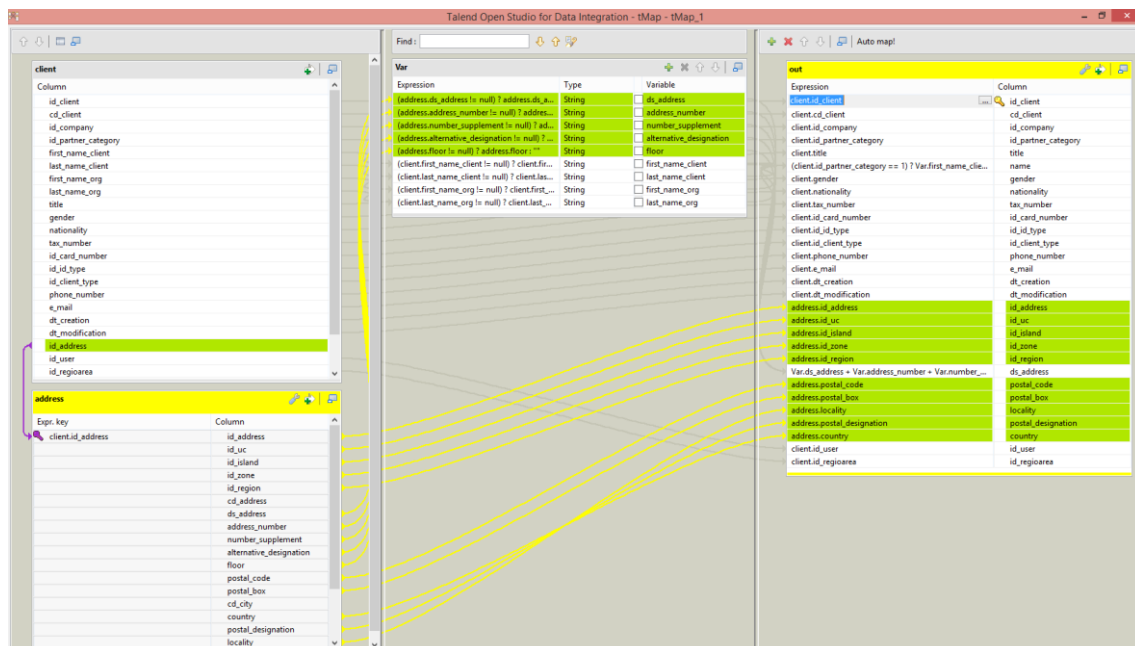


Figura 3.19 Mapeamento de campos na fase II de transformação

Na Figura 3.19 encontra-se o mapeamento concretizado para a fase II de transformação, onde se destaca a região central que possibilita a concretização de um conjunto de verificações e processamento dos dados. No presente caso, esta funcionalidade do componente tMap (descrita na Subsecção 3.3.3) foi bastante útil para verificar se alguns campos se encontravam a *null* (levando à ocorrência de erros nos *jobs*) e facilitar a concatenação dos dados de vários campos da tabela de Clientes e de Moradas. Esta concatenação, para além de ser necessária por motivos de apresentação nos relatórios operacionais, deveu-se ao facto de os nomes de clientes e das descrições das moradas estarem repartidos por múltiplos campos das tabelas. A título de exemplo, relativamente aos Clientes, o primeiro e último nome do cliente/organização estão divididos em dois campos que foi necessário agregar para que o seu nome fosse apresentado de forma inteligível nos relatórios. No entanto, este não foi caso único de utilização desta funcionalidade do tMap. Dada a integração do Talend com o Java, as variáveis do tMap permitem a inclusão de código Java que facilita a conversão de tipos de dados, o tratamento de *strings* e o cálculo entre campos e variáveis, também concretizados nalguns *jobs*.

Após o término dos *jobs* referentes a esta fase II de transformação (cerca de 19), os dados ficaram prontos a ser utilizados pelo Pentaho, na tarefa de construção/desenho dos relatórios.

3.3.5 Monitorização de erros

A ocorrência de erros, que podem advir, por exemplo, de falhas na conexão ao SAP IS-U, alterações do tipo de dados extraídos/transformados ou falhas no servidor em que está alojado o SGC, durante a execução dos *jobs* é uma possibilidade. Então, para possibilitar a obtenção de informações relativas a erros ocorridos durante a execução dos *jobs*, implementou-se um *subjob* em todos os *jobs*. Tal como se pode constatar na Figura 3.20, os componentes que constituem este *subjob* são:

- tLogCatcher: após ter sido despoletado por uma exceção de Java, guarda um conjunto de campos e mensagens relativos à ocorrência da mesma.
- tMap: transforma e mapeia dados da(s) fonte(s) para o(s) destinatário(s).
- tPostgresqlRow: executa uma *query* na base de dados.

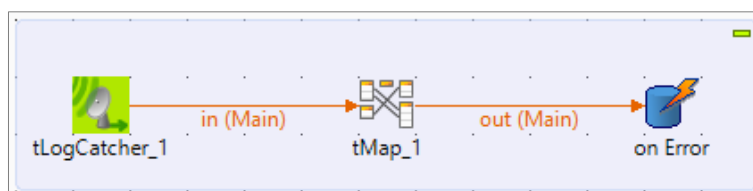


Figura 3.20 Subjob de controlo de erros

Aquando da ocorrência de uma exceção de Java durante a execução normal de um *job*, o componente tLogCatcher é ativado, selecionando-se os campos relevantes em tMap, e o tPostgresqlRow concretiza a *query* que desencadeia a execução da função *on_job_error* na *data staging area*. Esta função atualiza a tabela de controlo de execução de *jobs*, onde são registados, entre outros dados, as horas de início e fim, o nome, o estado (*Running/Complete/Error*) e as descrições de erros.

Para além da função *on_job_error*, existem mais duas responsáveis pela atualização da referida tabela de controlo (ver Apêndice III): i) *on_job_begin*, que insere uma nova linha de registo a cada início de um *job*; e ii) *on_job_end*, que atualiza a linha inserida após o término da execução do *job*.

3.3.6 Otimização do desempenho

A execução do processo de ETL durante a janela de oportunidade (espaço de tempo disponível, por norma reduzido, para a execução das três fases do processo) é de extrema importância. Ao longo do desenvolvimento dos processos do Talend, foi possível constatar que o tempo de execução dos *jobs* relativos aos dados-mestre e dados transacionais era muito elevado (podendo demorar muitas horas) devido à quantidade de dados envolvidos. Mais, para além de se prever o aumento do tempo de execução com o crescimento do volume de dados, esta questão verificava-se tanto na fase de extração como na de transformação dos dados.

Foram efetuados alguns testes, na mesma máquina, tentando simular as mesmas condições para cada *job*. Utilizou-se uma máquina com sistema operativo Linux, processador *octa-core*, memória RAM de 32 GB e disco rígido de 400 GB.

Registaram-se os dados de execução dos *jobs* de extração relativos a Clientes, Moradas, Locais de Consumo e Prédios, os quais são apresentados na Tabela 3.1:

Processos	Número de registos	Duração da execução	Velocidade média da execução
Clientes	78 522	4370,84 s	17,96 registos/s
Moradas	313 523	15813,24 s	19,83 registos/s
Prédios	114 691	5922,48 s	19,37 registos/s
Locais de Consumo	143 455	7312,88 s	19,62 registos/s

Tabela 3.1 Dados relativos à execução dos *jobs* de extração

Por forma a evitar a sobrecarga do SGC, a execução dos *jobs* do processo de ETL deve ocorrer em horas de menor utilização do mesmo, o que torna o tempo de execução do ETL um ponto crítico pois o processo de extração tem de estar concluído num período limitado. Para que os relatórios operacionais estejam o mais atualizados possível (dados referentes à última jornada laboral), é também necessário que o processo de transformação esteja finalizado. No entanto, não sendo possível paralelizar, o início de alguns *jobs* depende da conclusão de outros e o progresso da etapa de transformação está sujeito à duração dos mesmos, o que rapidamente coloca problemas de escala. Assim, foi sugerido que se procurasse melhorar o desempenho dos *jobs* de extração e transformação de dados-mestre e dados transacionais.

Após alguma investigação, desenvolveu-se um processo alternativo que passa pelo carregamento de dados em massa e pela utilização de ficheiros. Em [15], os autores apresentam um conjunto de estratégias e dicas de otimização do carregamento massivo de dados, bem como um estudo experimental que permite constatar que o carregamento *bulk* (em massa) revela o melhor desempenho.

No decorrer da arquitetura dos *jobs*, verificou-se que um dos fatores que contribuía para a elevada duração de um processo era a inserção registo-a-registo efetuada pelo componente *tPostgresqlOutput*. Assim, a alternativa passou por usar dois componentes (*tPostgresqlOutputBulk* e *tPostgresqlBulkExec*) que permitem o carregamento de dados em massa e cujo funcionamento exige que os dados de entrada provenham de um ficheiro.

Relativamente à extração de dados, de modo a possibilitar a sua realização através de ficheiros, a equipa técnica de SAP alterou a função RFC que tinham desenvolvido para que o resultado da sua invocação fosse um ficheiro de texto. Esta abordagem difere da inicialmente pensada no volume de dados extraídos para ficheiros de texto, na qual se previa que todos os dias seriam exportados vários ficheiros com todos os campos de todas as tabelas que incluíssem informação necessária aos relatórios.

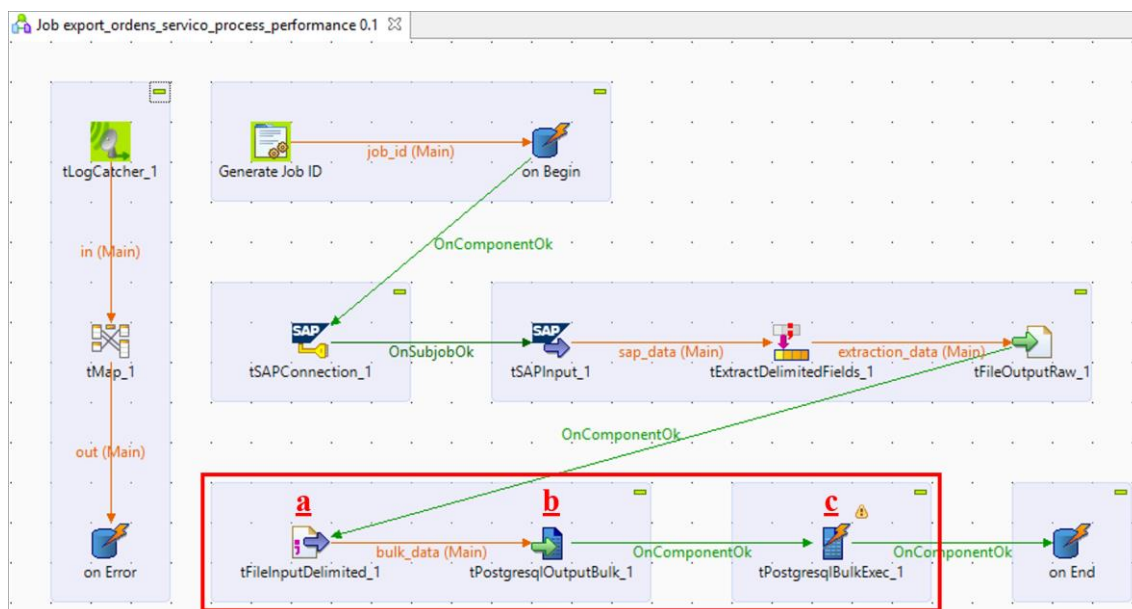


Figura 3.21 Job de extração de dados modificado

Após esta adaptação, o fluxo de operações foi remodelado, tal como se pode constatar na Figura 3.21. A principal alteração (destacada a vermelho) visa: a) a leitura de um ficheiro com os dados extraídos do SGC; b) a escrita de um ficheiro de carregamento de dados específico do PostgreSQL; e c) o carregamento em massa dos dados para a base de dados.

Processos	Número de registos	Duração da execução	Velocidade média da execução
Clientes	78 606	1,63 s	48 224,54 registos/s
Moradas	314 083	4,99 s	62 942,48 registos/s
Prédios	114 759	1,18 s	97 253,39 registos/s
Locais de Consumo	143 661	1,55 s	92 684,52 registos/s

Tabela 3.2 Resultados obtidos após a modificação dos jobs de extração

Esta nova abordagem permitiu reduzir o tempo de extração em três ordens de grandeza, tal como se pode constatar pelos resultados obtidos, detalhados na Tabela 3.2¹⁵ acima (os resultados antes da modificação estão na Tabela 3.1).

	Processos	Número de registos	Duração da execução	Velocidade média da execução
Transformação I	Clientes	78 606	17 340,22 s	4,29 registos/s
	Moradas	314 083	77 883,72 s	4,03 registos/s
	Prédios	114 759	26 153,47 s	4,39 registos/s
	Locais de Consumo	143 661	33 951,51 s	4,23 registos/s
Transformação II	Clientes	78 606	8 534,23 s	9,21 registos/s
	Moradas		Não aplicável ¹⁶	
	Prédios	114 759	12 390,40 s	9,26 registos/s
	Locais de Consumo	143 661	15 480,02 s	9,28 registos/s

Tabela 3.3 Dados relativos à execução dos *jobs* de transformação

	Processos	Número de registos	Duração da execução	Velocidade média da execução
Transformação I	Clientes	78 606	8,91 s	8 822,22 registos/s
	Moradas	314 083	11,32 s	27 745,85 registos/s
	Prédios	114 759	7,84 s	14 637,63 registos/s
	Locais de Consumo	143 661	9,86 s	14 570,08 registos/s
Transformação II	Clientes	78 606	8,68 s	9 055,99 registos/s
	Moradas		Não aplicável ¹⁶	
	Prédios	114 759	8,14 s	14 098,16 registos/s
	Locais de Consumo	143 661	10,38 s	13 840,17 registos/s

Tabela 3.4 Resultados obtidos após a modificação dos *jobs* de transformação

¹⁵ Os valores referentes ao número de registos diferem ligeiramente (cerca de 0,1%) dos apresentados na Tabela 3.1 uma vez que, com o decorrer do projeto, foram criados novos registos e a execução dos *jobs* foi concretizada em diferentes momentos.

¹⁶ Como não existe relatório de moradas, estes dados não são sujeitos à fase II de transformação.

Quanto aos *jobs* responsáveis pelas fases de transformação, a alteração realizada foi idêntica. Após a transformação dos dados, ao invés de se inserir registro-a-registo na base de dados, escreve-se para um ficheiro com os campos delimitados, que é depois utilizado nos *subjobs* responsáveis pelo carregamento em massa descrito para a fase de extração. As tabelas acima detalham os dados de execução antes (Tabela 3.3) e depois (Tabela 3.4) da modificação dos *jobs*. Como se pode constatar pelas tabelas acima, os resultados obtidos foram muito positivos, permitindo reduzir significativamente a duração dos processos.

Na Figura 3.22 apresenta-se um esquema que resume o fluxo de trabalho do processo de ETL.

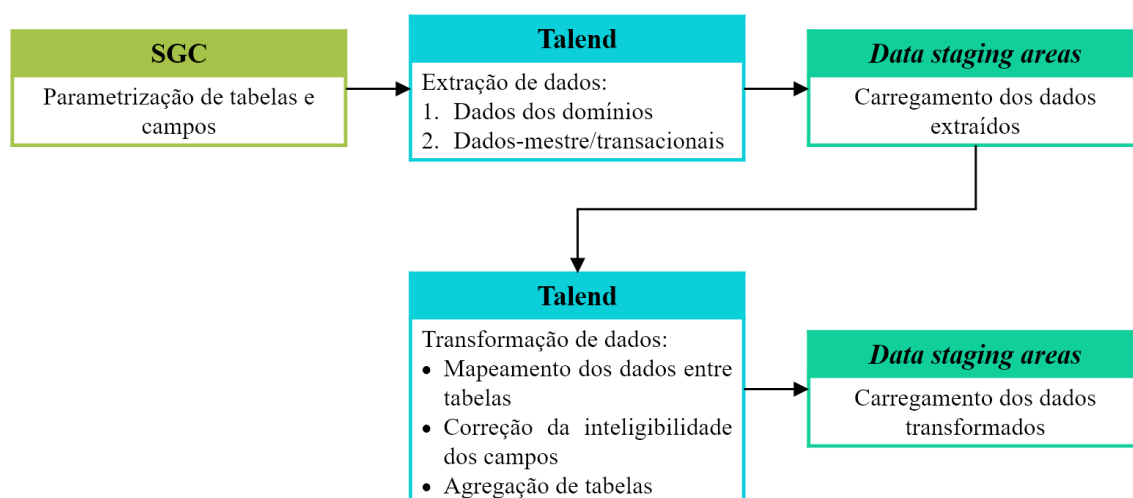


Figura 3.22 Visão geral do processo de ETL

3.3.7 Validação do trabalho realizado

A validação do processo de ETL centrou-se na melhoria dos tempos de execução, uma vez que a janela de tempo disponível para a conclusão deste processo é limitada. Os resultados obtidos revelaram que os *jobs* desenvolvidos dão resposta às diferentes fases de um processo de ETL, em tempo útil, tendo reduzido o tempo de execução do processo.

Para além da melhoria dos tempos de execução, a validação dos *jobs* passou também pelos dados resultantes do processo. A equipa funcional e o cliente têm vindo a validar os campos obtidos, bem como a conformidade dos mesmos com os que estão no SGC. No entanto, este processo de validação encontra-se estagnado devido a divergências financeiras com o cliente (não relacionadas com os relatórios).

3.4 Relatórios operacionais

A partir do momento em que se começaram a obter dados, foi possível concretizar os relatórios operacionais (listados na Secção 3.2) no Pentaho. Por não depender da

existência de dados e por se conhecerem os campos necessários, foi possível avançar com o desenho dos relatórios, durante o desenvolvimento do processo de ETL. Na Figura 3.23 encontra-se o resultado final de um relatório operacional.

Figura 3.23 Exemplo de um relatório operacional (Planos de Pagamento)

3.4.1 Análise de requisitos

O processo de realização dos relatórios foi idêntico para todos. Inicialmente, definiu-se o *layout* do relatório de acordo com os requisitos que são transversais a todos os relatórios, dos quais se destacam (Figura 3.24):

1. Inclusão dos logótipos do cliente;
2. Possibilidade de filtrar os dados através de um conjunto de filtros definidos para cada relatório;
3. Existência de um campo totalizador que exiba o número de linhas do relatório, antes ou após a aplicação de filtros, bem como campos calculados (por exemplo, percentagens, médias);
4. Todos os relatórios devem estar identificados com um título;
5. Os campos apresentados devem ter um cabeçalho que os identifique.

Figura 3.24 Elementos de um relatório operacional (Relatório de Dívidas)

Seguidamente, especificam-se em maior detalhe os requisitos identificados:

- **Funcionamento dos filtros:** filtragem de campos de texto independente da existência de acentos ou letras maiúsculas/minúsculas; intervalo de datas definido retorna apenas registos cujo campo filtrado pertence ao intervalo; registos são filtrados de acordo com a opção selecionada.
- **Conformidade dos relatórios face aos requisitos definidos:** campos, informação e *layout* apresentados nos relatórios têm de estar de acordo com aquilo que foi estipulado nos requisitos do projeto SGC.
- **Funcionamento dos totalizadores:** os valores dos campos totalizadores devem espelhar resultados corretos.
- **Interface dos relatórios:** os logótipos do cliente e nome do relatório devem existir em todos os relatórios; os cabeçalhos dos campos devem ter todos a mesma altura; a largura das colunas deve ser suficiente para se compreender a informação.
- **Conformidade dos dados com o SGC:** os dados apresentados nos relatórios devem corresponder aos dados presentes no SGC.
- **Exportação dos dados para outros formatos:** deve ser possível efetuar a exportação dos dados, filtrados ou não, para um ficheiro de Excel.
- **Nomenclatura apresentada:** nos casos em que se aplique, o cabeçalho da coluna deve apresentar a especificação da moeda; os campos numéricos devem efetuar a separação dos milhares e casas decimais.
- **Tempo de geração de relatórios:** qualquer relatório deve ter um tempo de geração adequado à utilização do mesmo, isto é, deve ser gerado rápido o suficiente para que o utilizador não considere que este não funciona ou que algo correu mal.

3.4.2 Implementação dos relatórios

Após o desenho da interface, foi necessário definir a conexão à base de dados para que o Pentaho pudesse aceder aos dados dos relatórios. Com esta conexão, foi possível concretizar um conjunto de interrogações SQL para apresentar os dados nos relatórios, como os que constam na Figura 3.23 sobre Planos de Pagamento, bem como para obter valores para preencher os filtros.

Na sequência da definição das *queries*, criou-se um conjunto de *parameters* para serem utilizados nos filtros. Apresenta-se na Figura 3.25 a interface de edição dos *parameters*, na qual se (1) nomeia o mesmo; (2) define-se a descrição que aparece na interface do relatório; (3) o tipo de dados que recebe para filtrar (inteiro nas *drop-down lists*, texto nas *text-boxes*, data nos *date pickers*); (4) o tipo de *prompt* (*drop-down*, *text-box*, *date picker*) disponibilizada pelo *parameter*; (5) qual a *query* que reúne os dados a apresentar no filtro; (6) o campo que servirá de filtro na *query*; e (7) o campo que será

apresentado ao utilizador. No caso dos domínios, compostos por identificador (id) e descrição (dsc), o id é o campo de filtragem e dsc é o campo apresentado ao utilizador.

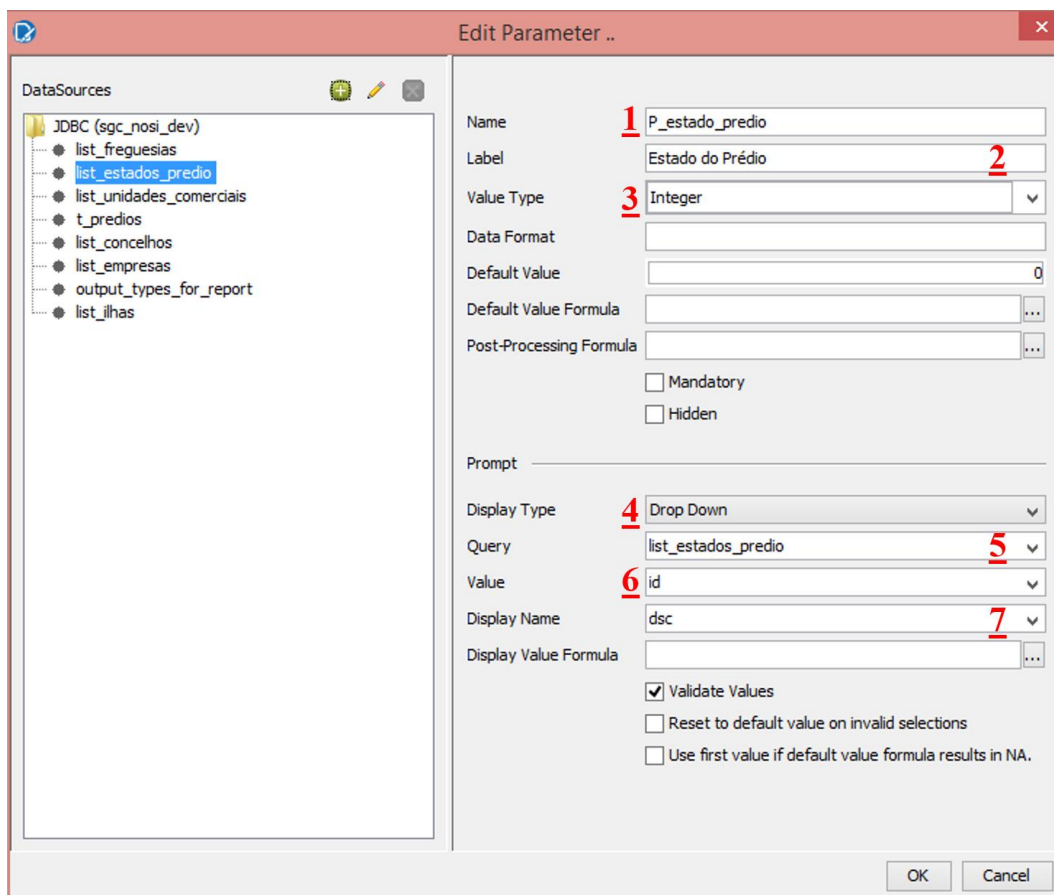


Figura 3.25 Interface de edição de *parameters* no Pentaho

Depois de os *parameters* estarem definidos para cada relatório, foi necessário incluí-los na *query* SQL de obtenção dos dados do respetivo relatório operacional. Como se pode visualizar na Figura 3.26, o SELECT realizado apresenta na condição WHERE alguns elementos identificados por um cifrão (\$). Estes elementos representam valores provenientes de *parameters*, especificamente da coluna definida no campo *Value* (elemento 6 da Figura 3.25). De acordo com a opção escolhida pelo utilizador, assim será preenchido o valor de \$(Nome_Do_Parameter).

Por exemplo, admita-se que o utilizador escolheu a opção “Devoluto” no filtro Estado do Prédio (*drop down list*), cujo id correspondente na tabela de estados do prédio é 3. A condição vai então ser avaliada como `id_estado_predio = 3`, filtrando todos os prédios de modo a obterem-se apenas os registos em que o `id_estado_predio` seja 3.

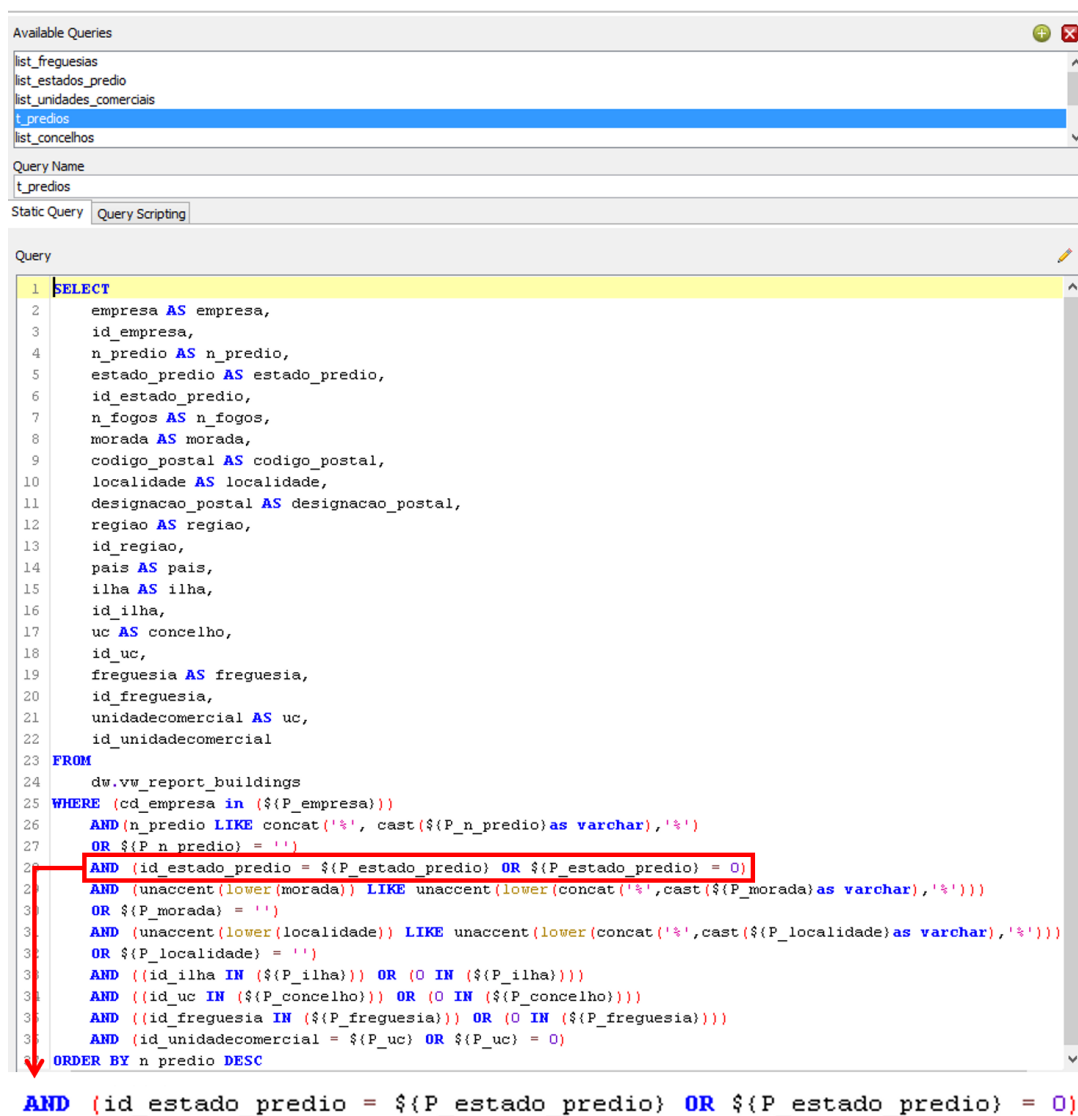


Figura 3.26 Interface de definição de *queries* do Pentaho (em cima) e detalhe da condição WHERE (em baixo)

Por fim, o relatório pode ser publicado no servidor fornecido pelo Pentaho, passando a estar acessível através de um *link*, o qual pode ser utilizado no UFE pelo cliente.

3.4.3 Problema de usabilidade e sua resolução

Relativamente a estes relatórios, levantou-se uma questão de usabilidade importante, nomeadamente quanto ao carregamento demorado dos dados pelo *browser*. Inicialmente, a submissão dos valores escolhidos nos filtros era automática, pois todos têm um valor por defeito que representa a não aplicação do filtro, e, portanto, a inicialização de um relatório subentendia a apresentação de todos os dados (sem filtros aplicados). No entanto, constatou-se que o Pentaho executa o carregamento de *todos* os registos de um relatório, independentemente do número de linhas que vai apresentar ao

utilizador, de cada vez que o relatório é inicializado ou até mesmo quando se muda de página. Esta questão é relevante quando o volume de dados é muito grande, tal como acontece, por exemplo, nos relatórios de Locais de Consumo (número de registos superior a cento e quarenta mil) ou de Cobranças (número de registos superior a cem mil), e em que o Pentaho demora muito tempo para apresentar qualquer resultado, deixando o utilizador à espera sem qualquer *feedback*.

Este problema foi analisado em parceria com um elemento da equipa de *analytics*, nomeadamente quanto à eficiência das *queries* que estão a ser feitas à base de dados dos relatórios operacionais, tendo-se verificado que a origem do problema não estava aí, mas sim no Pentaho. Para além do tempo de carregamento, cada relatório apresenta ainda um número considerável de colunas (sempre superior a 10) e linhas (em média, mais de 100 mil), cuja apresentação ao utilizador é demorada. Até ao momento, a solução adotada foi de não permitir o carregamento automático dos relatórios e obrigar a definição de um conjunto de filtros, que permite a redução à partida dos dados apresentados.

3.4.4 Validação do trabalho realizado

A validação dos relatórios operacionais foi concretizada por elementos da equipa funcional do SGC bem como por elementos do cliente. Os testes efetuados abrangeram a interface, o desempenho e os dados presentes nos relatórios, nomeadamente validando o cumprimento dos requisitos.

Este processo de validação foi facilitado pela plataforma JIRA (Atlassian), a qual permite o rastreamento e reportação de *bugs*/erros/falhas. Todos os intervenientes do projeto têm um perfil na plataforma, que lhes permite reportar *issues* a outros utilizadores e vice-versa, de acordo com o âmbito da sua atividade. Através desta plataforma, o processo de validação dos relatórios decorreu de forma ágil, permitindo a análise e correção de diversos *issues* identificados (cerca de noventa, em duas semanas).

De momento, os *issues* por resolver relacionam-se com limitações da versão do Pentaho utilizada nos relatórios operacionais (em versões mais recentes, algumas questões já foram resolvidas), bem como com a falta/incorreção de alguns campos de dados, cuja revisão por parte da equipa funcional do SGC está pendente. O requisito relativo ao tempo de geração foi contornado, mas não resolvido.

Importa ainda realçar dois pontos: i) os relatórios operacionais fizeram parte da formação que a equipa SGC realizou em Abril com o cliente; e ii) os modelos dos relatórios desenvolvidos foram utilizados num *Business BluePrint* (BBP) apresentado no contexto de um novo projeto para os Serviços Municipalizados de Água e Saneamento (SMAS) de Sintra.

3.5 Dashboards de BI

Os *dashboards* de BI permitem visualizar os dados de uma forma diferente da que é disponibilizada pelos relatórios operacionais. Muito embora os *dashboards* que foram criados não estejam relacionados com o SGC, também seria possível concretizar um conjunto de *dashboards* para o SGC com o modelo de dados desenvolvido para os relatórios operacionais.

No decorrer de uma formação que a equipa do SGC realizou em Cabo Verde, em abril, surgiu a oportunidade de criar um conjunto de *dashboards* para serem incluídos em duas provas de conceito, uma para um cliente da Zâmbia e outra para um cliente do Gana. Uma vez que se tratou de uma prova de conceitos, não houve recolha de requisitos.

Na Figura 3.27, apresentam-se as etapas do processo de desenvolvimento dos *dashboards*, onde se destacam os elementos com asterisco (*), que representam tarefas realizadas por outras pessoas.

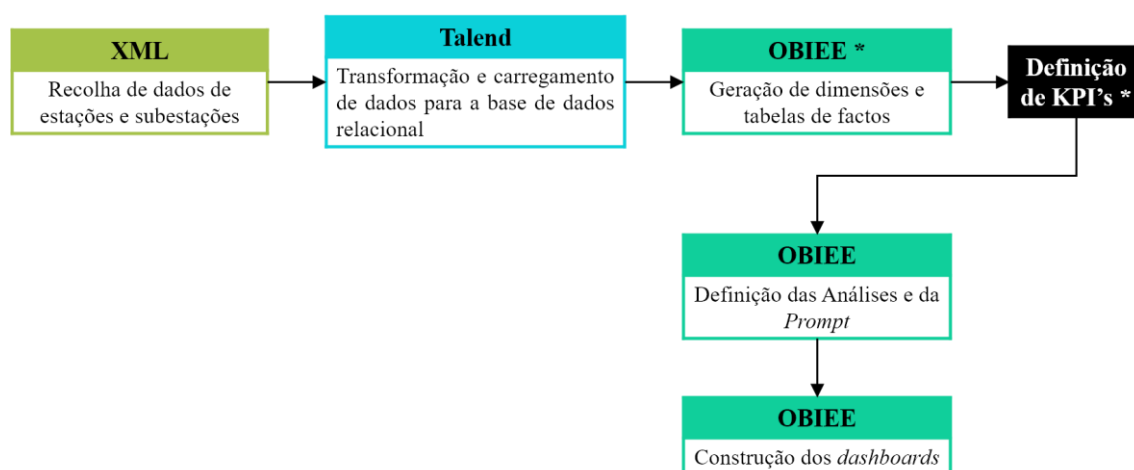


Figura 3.27 Etapas do processo de desenvolvimento dos *dashboards*

Foram concretizados três *dashboards* em OBIEE: i) *Strategic Indicators*, que faculta um conjunto de indicadores estratégicos com base nos objetivos da empresa para os próximos anos (Figura 3.31); ii) *Operations*, que fornece dados analíticos sobre a atividade operacional da empresa (Figura 3.32); e iii) *Assets*, que apresenta informação relativa aos ativos (Figura 3.33).

3.5.1 Análise preliminar

Por forma a conhecer e compreender a realidade do sector energético em ambos os países, foram analisados um conjunto de relatórios executivos produzidos pelas respetivas empresas de energia da Zâmbia e do Gana. Esta análise permitiu a recolha de dados relativos à produção de energia em ambos os países nos últimos anos, bem como as metas que se pretendem alcançar. Foi também utilizado um ficheiro eXtensible

Markup Language (XML) com informações sobre as estações e subestações da Zâmbia, as quais foram obtidas através de um *job* do Talend, que transformou e carregou os dados correspondentes para uma base de dados relacional.

Enquanto um elemento da equipa de *analytics* gerou as dimensões e as tabelas de factos no OBIEE, um grupo de dois outros elementos dessa equipa definiu um conjunto de *Key Performance Indicators* (KPIs). Os KPIs definidos foram:

- **Strategic Indicators:** i) Produção de energia no presente ano e objetivos de produção para 2030; ii) Quilómetros de transmissão de energia no ano atual e objetivos para 2030; iii) Distribuição de energia e serviços ao consumidor para este ano e objetivos para 2030; iv) Variação da tendência de importação de energia, exportação de energia e consumo *per capita*.
- **Operations:** i) Disponibilidade do transporte de energia nos últimos seis meses e respetiva variação; ii) Distribuição da energia produzida nos dois semestres do ano e variação; iii) Disponibilidade de distribuição da energia ao longo de seis meses e qual a variação; iv) Produção de energia ao longo de um ano face à média anual.
- **Assets:** i) Produção de energia por fonte de energia; ii) Energia produzida em meses consecutivos e sua variação; iii) Produção de energia ao longo de um ano; iv) Energia gerada por subestação, no mês atual e no anterior.

A equipa analisou o leque de possibilidades de gráficos do OBIEE e concluiu que aqueles que melhor serviam os *dashboards* eram os gráficos de barras e os gráficos *donut/semicircle donuts*. No caso das tendências, utilizaram-se ainda gráficos de linhas, que demonstram melhor a sua evolução ao longo do tempo.

Na sequência da questão dos tipos de gráficos a incluir, optou-se pela utilização de uma biblioteca de *JavaScript*, o D3.js¹⁷, que possibilitou a inclusão de gráficos visualmente mais atrativos e funcionais que os do OBIEE (nomeadamente, os gráficos *donut* e *semicircle donut*). No entanto, a utilização de *JavaScript* acarreta também a necessidade de as tecnologias terem de estar ligadas à Web, o que pode ser uma limitação.

3.5.2 Construção dos *dashboards*

Depois de a geração das dimensões e das tabelas de factos estarem concluídas, foi possível passar à construção dos *dashboards* e das respetivas análises. No contexto do OBIEE, um *dashboard* é composto por uma ou mais análises e uma mesma análise pode ser usada em múltiplos *dashboards*.

¹⁷ <https://d3js.org/>

A criação de uma análise passou por definir quais as dimensões e medidas que fazem parte da mesma (separador Critérios, na Figura 3.28) e pela concretização da forma de visualização dos dados (separador Resultados). Inicialmente foi ainda criada uma *prompt* que possibilita a filtragem dos dados segundo determinados campos (ano, mês, província e distrito). Também as *prompts* podem ser utilizadas em mais do que um *dashboard*, tal como acontece nos relatórios de *Operations* (ver Figura 3.32) e *Assets* (ver Figura 3.33).

Na Figura 3.28 encontram-se detalhadas as áreas de relevo no separador Critérios, em que: a) apresenta as dimensões, tabelas de factos e medidas disponíveis; b) mostra as dimensões e medidas a utilizar (*drag-and-drop* a partir do Repositório); e em c) seleccionam-se os dados a serem filtrados pela *prompt*.

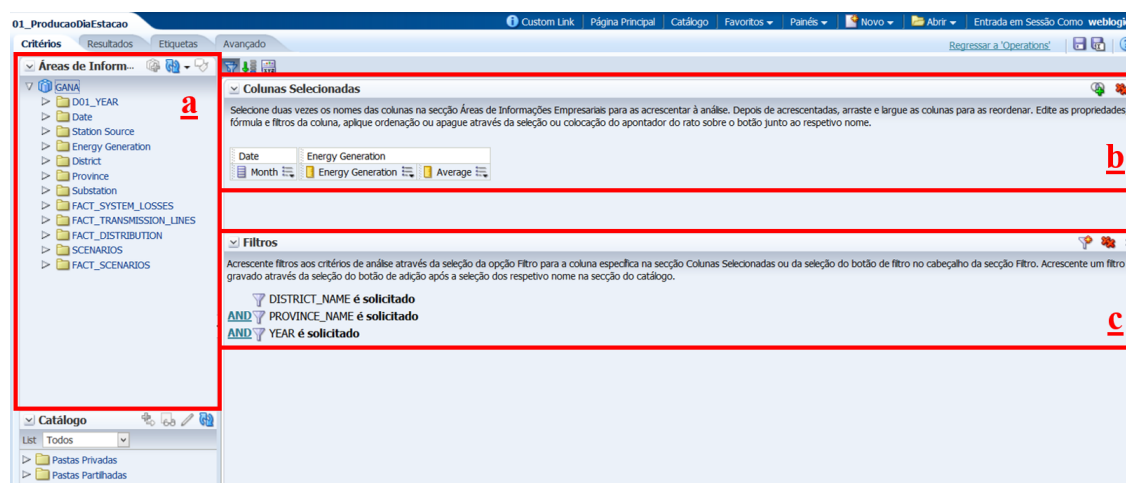


Figura 3.28 Interface de edição de uma análise do OBIEE (separador Critérios)

Seguidamente, determinou-se a forma de visualização dos dados. Os tipos de visualizações de dados utilizados foram os gráficos de barras, de linhas, *donuts*, *semicircle donuts* e narrativas (dados textuais).

A Figura 3.29 ilustra a concretização de um gráfico *donut*, utilizando a biblioteca D3.js, cuja inclusão foi possível através da componente Narrativa do OBIEE. Em a) declara-se a variável do tipo *array* que conterá os dados da análise e define-se o estilo da secção (*div*) em que ficará o gráfico; em b) inserem-se os dados da análise no *array*, em que @1 e @2 são substituídos pelos dados que compõem a análise (neste caso, a fonte de energia e a quantidade de energia gerada); em c) inclui-se o código de geração do gráfico, no qual se chama o *array* na respetiva função, definem-se as suas cores, a legenda, o tamanho do gráfico, entre outros; e em d) é apresentada uma pré-visualização do gráfico que atualiza à medida que são feitas alterações no código.

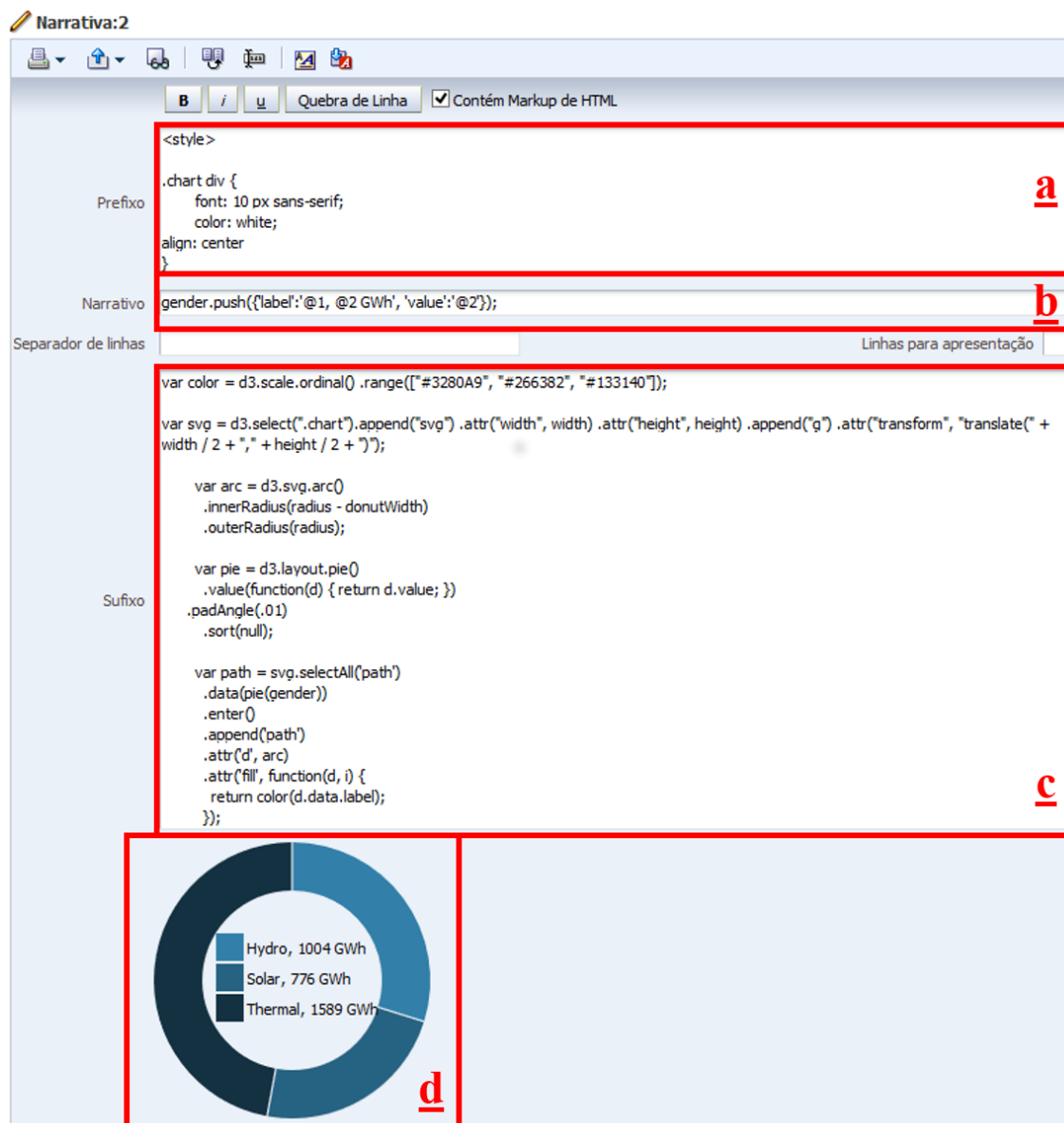


Figura 3.29 Definição do tipo de visualização – gráfico donut

Após a conclusão das análises e da *prompt*, pôde-se compor os *dashboards*. A Figura 3.30 ilustra a disposição dos diversos elementos do *dashboard Operations*, nomeadamente: 1) a caixa de texto que apresenta o título; 2) a *prompt* de filtragem dos dados; e 3) múltiplas análises aos dados (seis na zona superior e uma ocupando toda a zona inferior). De um *dashboard* para outro, o OBIEE permite a reutilização das análises, bem como das *prompts* definidas, evitando a repetição de trabalho.

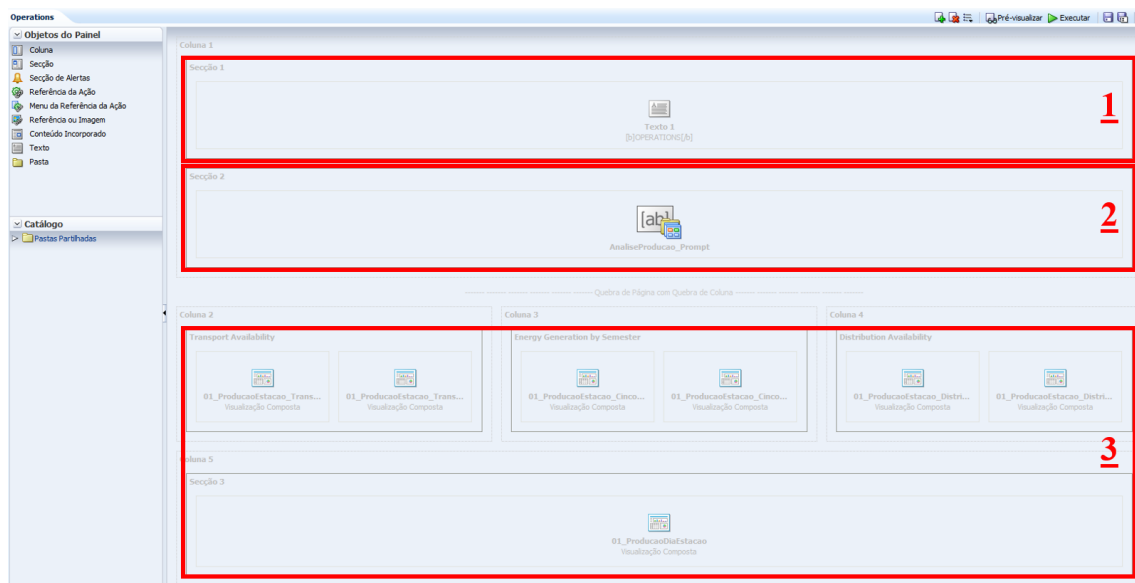


Figura 3.30 Interface de edição de um *dashboard* do OBIEE

De seguida, apresentam-se os resultados finais da construção dos *dashboards*. Como se pode verificar, o *dashboard* da Figura 3.31 não tem *prompt* (conjunto de campos que estão no topo da Figura 3.32 e que permitem a aplicação de filtros). Também os gráficos utilizados diferem entre os *dashboards*. Na figura abaixo encontram-se i) gráficos *semicircle* (no topo), que permitem comparar os objetivos da empresa nos próximos anos; ii) gráficos de barras, que permitem visualizar a evolução das métricas ao longo de um período de tempo; e iii) gráficos de linhas, que descrevem tendências. Na Figura 3.32, para além dos gráficos de barras laterais, existe também um iv) gráfico *donut*, que apresenta a distribuição da energia produzida por semestre; e um v) gráfico na parte inferior do *dashboard*, que apresenta a produção de energia em cada mês e que facilita a visualização dos meses mais e menos produtivos, por exemplo.



Figura 3.31 *Dashboard* relativo a *Strategic Indicators*

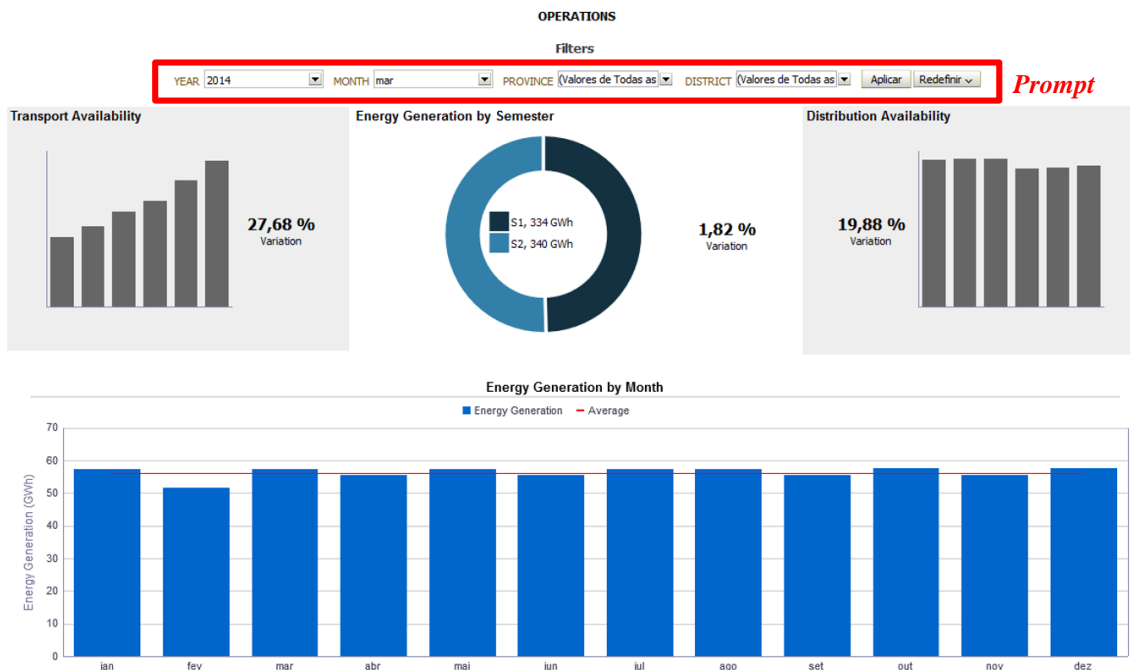


Figura 3.32 Dashboard relativo a Operations

Na Figura 3.33, para além do gráfico *donut* (à esquerda) e do gráfico de barras (à direita), existe ainda uma narrativa (dados textuais) no centro e um gráfico de barras (na parte inferior) que permite comparar a produção de energia do mês atual e do mês anterior, em cada estação.

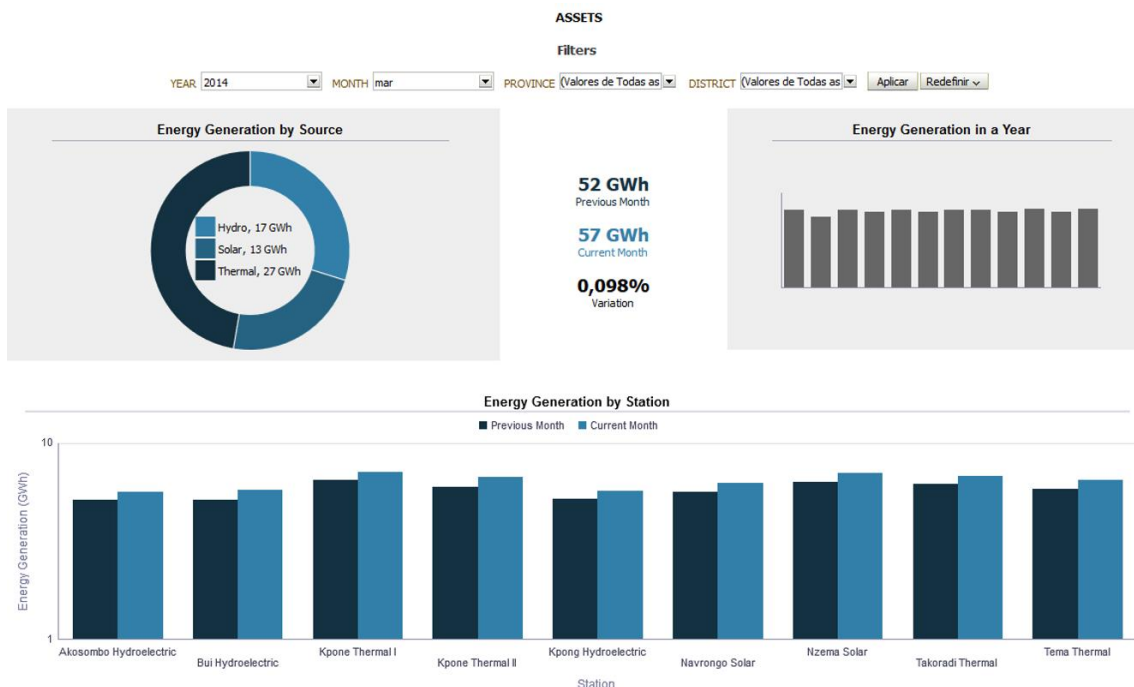


Figura 3.33 Dashboard relativo a Assets

3.5.3 Validação do trabalho realizado

Os *dashboards* foram avaliados por elementos da equipa de *analytics*, tendo em conta a informação que estes disponibilizam e a forma como esta informação se apresenta ao utilizador. Uma vez que foram definidos múltiplos KPIs, verificou-se que estes eram corretamente apresentados nos *dashboards* e testou-se o funcionamento dos filtros, que devem manter a coerência da informação detalhada nos diferentes gráficos. Na prática, a validação destes *dashboards* ocorreu nas reuniões formais já previstas, ou de forma mais livre, quando necessário, garantindo que os resultados finais forneciam a informação pretendida.

Para além desta validação por parte da equipa de *analytics*, responsável pela definição das provas de conceito, aguarda-se ainda *feedback* sobre as funcionalidades apresentadas (nas quais se incluem os *dashboards*) por parte dos clientes da Zâmbia e do Gana.

3.6 Sumário

Neste capítulo apresentou-se o ambiente de trabalho em que decorreu o estágio, incluindo as ferramentas utilizadas, a organização do trabalho da equipa e o processo de desenvolvimento de *software* adotado pela equipa.

Seguidamente, detalhou-se o modelo de dados do SGC, descreveu-se pormenorizadamente o processo de ETL desenvolvido, bem como a otimização concretizada e os resultados obtidos com esta ao nível do carregamento de dados. Depois, explicitou-se a implementação dos relatórios operacionais através do Pentaho e a construção dos *dashboards* com o OBIEE.

No próximo capítulo, discutem-se as conclusões alcançadas, identificam-se as principais contribuições e as dificuldades encontradas no decorrer do PEI, bem como o trabalho futuro.

Capítulo 4

Conclusão

De seguida, descrevem-se as principais contribuições do presente Projeto de Engenharia Informática (PEI), as competências adquiridas no decorrer do estágio, as principais dificuldades encontradas bem como possíveis melhorias e desenvolvimentos futuros.

4.1 Principais contribuições

Inicialmente, não existia qualquer modelo de dados do Sistema de Gestão Comercial (SGC), levando a que a compreensão do sistema por parte de alguém novo ao projeto fosse difícil. Assim, uma das contribuições do PEI foi o modelo de dados concretizado, o qual permite obter uma visão abrangente dos principais conceitos do SGC bem como das relações entre os mesmos. Este modelo tem também um papel importante na documentação relativa ao processo de *Extraction-Transformation-Loading* (ETL) e aos relatórios, facilitando a passagem de conhecimento e garantindo boas bases de partida para o trabalho futuro.

Outra contribuição foi o processo de ETL desenvolvido (objetivo O1). A execução dos *jobs* era muito demorada, impossibilitando que os dados fossem extraídos/tratados em tempo útil. Por este motivo, as fases de extração e transformação foram otimizadas para que os dados apresentados nos relatórios (e consequentemente ao cliente) estejam o mais atualizados possível, garantindo que ambas as fases ocorrem na janela de oportunidade, sem interferir no correto funcionamento do SGC. Também a conformidade dos dados dos relatórios operacionais foi alcançada.

Outro objetivo do trabalho foi o desenho e concretização de relatórios operacionais (objetivo O2). Os relatórios são funcionais, cumprem os requisitos do cliente e permitem a concretização dos seus objetivos primários. Estes relatórios tiveram uma importante contribuição no SGC pois foram sempre alvo de elevado interesse por parte do cliente, tendo feito parte de uma formação concretizada em abril.

Por fim, os *dashboards* produzidos (objetivo O3) para os clientes do Gana e da Zâmbia fizeram parte das provas de conceito apresentadas, fornecendo um conjunto diferenciador de visualizações de indicadores de desempenho relevantes para os (possíveis) clientes.

4.2 Competências adquiridas

Este PEI permitiu a utilização de novas e variadas tecnologias, que possibilitaram a concretização dos principais objetivos definidos.

Relativamente ao processo de ETL, foram adquiridas competências relativas ao tratamento de dados, fluxos de informação e encadeamento lógico de operações. A concretização deste objetivo foi muito positiva pois permitiu que o desenvolvimento de um processo deste género fosse feito de forma diferente de outras anteriores, nomeadamente numa cadeira do ano curricular. Este contacto com uma ferramenta para implementar o processo de ETL permitiu obter uma visão diferente do processo e do fluxo de operações que dele fazem parte. Algumas competências técnicas foram também melhoradas, principalmente no que diz respeito à modelação de bases de dados e linguagem *Procedural Language/PostgreSQL* (PL/pgSQL).

Também foram adquiridas competências de *Business Intelligence* (BI) e *Analytics* através da concretização dos relatórios operacionais e *dashboards*. Estes dois tipos de relatórios de apoio à decisão surgem em vertentes que se complementam. Especificamente sobre os *dashboards*, estes possibilitaram a interação com uma ferramenta nova e a obtenção de um conjunto de competências da área de *Analytics*, nomeadamente na definição de indicadores de desempenho e na forma como se pode apresentar a mesma informação ao utilizador.

Por fim, este estágio possibilitou a aquisição de um conjunto de competências específicas do trabalho em empresa, nomeadamente o funcionamento numa equipa Scrum e a organização de tarefas. Considera-se que este aspeto foi muito positivo por ter fornecido uma visão do mundo empresarial e do trabalho em equipa muito particular, bem como a integração numa equipa multidisciplinar de pessoas com diferentes níveis de experiência e conhecimentos.

4.3 Principais dificuldades

Inicialmente, a maior dificuldade foi a não existência de um modelo de dados da base de dados que servia os relatórios. Seria expectável que este modelo tivesse sido concretizado antes da base de dados, mas não existia. Para alguém novo no projeto e desconhecedor do trabalho desenvolvido, a compreensão de todos os conceitos envolvidos num sistema da dimensão do SGC não se apresentou simples. No entanto,

esta falta motivou o desenho de um modelo de dados, tendo permitido um maior envolvimento com a equipa e com os objetivos do trabalho.

O desenvolvimento do processo de ETL e dos relatórios também esteve limitado por um outro fator: os relatórios, independentemente de serem um requisito do cliente, tiveram sempre pouca prioridade para os outros elementos da equipa, sejam eles técnicos ou funcionais. Esta questão foi uma das que exigiu maior esforço por dois motivos: i) uma boa parte dos processos ainda não estavam desenvolvidos, impossibilitando a extração de dados para os relatórios; e ii) embora não fossem uma necessidade para os outros elementos da equipa, faziam parte dos requisitos do cliente e tinham que ser desenvolvidos.

Quanto aos relatórios operacionais, houve dois que não foram concretizados pois, dada a sua complexidade, requerem desenvolvimentos por parte da equipa técnica do SGC. Inicialmente, pensou-se que poderiam ser obtidos tal como os restantes, mas com o decorrer do projeto verificou-se que não seria possível.

Por fim, uma outra dificuldade ao longo do PEI foi a resolução do carregamento lento dos dados para o *browser*. Nalguns casos, os relatórios demoravam mais de dez minutos a carregar, o que é inaceitável. A solução arranjada, de só mostrar dados quando todos os filtros tiverem sido aplicados, não é a ideal, mas tem possibilitado um melhor funcionamento dos relatórios.

4.4 Trabalho futuro

Quanto a trabalho futuro, seria importante concretizar os dois relatórios em falta, bem como os *jobs* responsáveis pela obtenção e tratamento dos mesmos. Falta também concluir o processo de validação dos relatórios, ainda a decorrer devido a divergências financeiras com o cliente.

Um outro aspeto que deve ser revisto é a solução adotada para o carregamento dos dados dos relatórios para o *browser*. Uma vez que o Pentaho tem as suas limitações, sugere-se a definição de um conjunto de campos obrigatórios, que filtrem os dados e impossibilitem o carregamento massivo e demorado dos dados.

Para finalizar, propõe-se que o modelo de dados desenvolvido se mantenha atualizado, como forma de documentação.

Bibliografia

- [1] Cisco, “Big Data: Grande Volume de Dados, Grande Potencial, Grande Prioridade - Cisco,” Cisco, [Online]. Disponível em: http://www.cisco.com/c/pt_pt/about/press/news-archive-2013/20130401.html. [Acedido em 31 Maio 2016].
- [2] M. Nutley, “Analysing The Challenges Facing Customer Insight,” Adobe Systems, [Online]. Disponível em: http://www.cmo.com/articles/2013/7/17/under_resourced_unde.html. [Acedido em 17 Dezembro 2015].
- [3] Microsoft, “Small Business BI: Understanding Customer Information | Microsoft SMB,” Microsoft, [Online]. Disponível em: <https://www.microsoft.com/en-us/business/business-news/business-intelligence/>. [Acedido em 17 Dezembro 2015].
- [4] Gartner, Inc., “Gartner Predicts Business Intelligence and Analytics Will Remain Top Focus For CIOs Through 2017,” Gartner, Inc., [Online]. Disponível em: <http://www.gartner.com/newsroom/id/2637615>. [Acedido em 17 Dezembro 2015].
- [5] Novabase, *Business Blueprint, Sistema de Gestão Comercial*, Lisboa, 2015.
- [6] R. Kimball e M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, Wiley, 2013.
- [7] H. J. Watson, G. Houdeshel e R. K. Rainer, *Building Executive Information Systems and Other Decision Support Applications*, Wiley, 1997.
- [8] D. J. Power, “A Brief History of Decision Support Systems,” DDSSResources.COM, [Online]. Disponível em: <http://dssresources.com/history/dsshhistory.html>. [Acedido em 12 Dezembro 2016].
- [9] C. Sezões, J. Oliveira e M. Baptista, *Business Intelligence*, SPI – Sociedade Portuguesa de Inovação, 2006.
- [10] J. Han, M. Kamber e J. Pei, *Data Mining: Concepts and Techniques*, Elsevier/Morgan Kaufmann, 2012.
- [11] E. Turban, R. Sharda, D. Delen e D. King, *Business Intelligence*, 2ª ed., Prentice Hall, 2011.

- [12] T. A. Majchrzak, T. Jansen e H. Kuchen, “Efficiency Evaluation of Open Source ETL Tools,” *Proceedings of the 2011 ACM Symposium on Applied Computing (SAC '11)*, pp. 287-294, 2011.
- [13] A. Simitsis , P. Vassiliadis e T. Sellis, “Optimizing ETL Processes in Data Warehouses,” *Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, pp. 564-575, 2005.
- [14] X. Liu e N. Iftikhar, “An ETL Optimization Framework Using Partitioning and Parallelization,” *Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC '15)*, pp. 1015-1022, 2015.
- [15] A. Vilaça e J. Abreu, *Estratégias Para o Carregamento Massivo de Dados Num Data Warehouse*, 2009.
- [16] P. S. Azevedo e C. Azevedo, “Os ERP's (Enterprise Resource Planning) Como Soluções Integradas Para a Indústria da Hotelaria e Turismo,” *Revista Portuguesa de Sistemas de Informação*, nº 14, pp. 7-13, 2001.
- [17] E. J. Umble, R. R. Haft e M. M. Umble, “Enterprise Resource Planning: Implementation Procedures and Critical Success Factors,” *European Journal of Operational Research*, vol. 146, nº 2, pp. 241-257, 2003.
- [18] V. Kale, *Implementing SAP™ R/3: The Guide for Business and Technology Managers*, 1ª ed., Sams Publishing, 2000, pp. 90-91.
- [19] Y. Yusuf, A. Gunasekaran e M. S. Abthorpe, “Enterprise Information Systems Project Implementation: A Case Study of ERP in Rolls-Royce,” *International Journal of Production Economics*, vol. 87, nº 3, pp. 251-266, 2004.
- [20] A. R. Gouveia, “Solução de Business Intelligence Para Seguros,” Dissertação de Mestrado, FCUP, 2013.
- [21] EmeraldCube Solutions, “Business Intelligence vs Operational Reporting vs Financial Reporting (Part 1),” [Online]. Disponível em: <http://www.emerald-cube.com/2013/09/ovr/>. [Acedido em 22 Junho 2016].
- [22] S. Few, *Information Dashboard Design*, O'Reilly, 2006.
- [23] The PostgreSQL Global Development Group, “PostgreSQL: About,” [Online]. Disponível em: <https://www.postgresql.org/about/>. [Acedido em 20 November 2015].
- [24] SAP SE, “SAP Utilities - SAP Library,” SAP SE, [Online]. Disponível em: http://help.sap.com/saphelp_utilities472/helpdata/en/c6/4dce68eafc11d18a030000e829fbbd/frameset.htm. [Acedido em 20 Novembro 2015].

- [25] M. James, “Scrum Reference Card | The Scrum Reference Card, and Other Articles by Michael James (MJ), Software Process Mentor,” [Online]. Disponível em: <http://scrumreferencecard.com/ScrumReferenceCard.pdf>. [Acedido em 15 Dezembro 2015].
- [26] R. S. Pressman, Software Engineering: A Practitioner’s Approach, McGraw-Hil, 2010.
- [27] SAP SE, “Domains - SAP Library,” SAP SE, [Online]. Disponível em: http://help.sap.com/saphelp_47x200/helpdata/en/cf/21ede5446011d189700000e8322d00/content.htm. [Acedido em 22 November 2015].
- [28] The PostgreSQL Global Development Group, “PostgreSQL: Documentation: 9.4: Schemas,” [Online]. Disponível em: <https://www.postgresql.org/docs/9.4/static/ddl-schemas.html>. [Acedido em 2016 Junho 23].

Apêndices

Apêndice I: Organização da equipa do SGC

Apêndice II: SQL dos principais objetos da base de dados

Apêndice III: SQL das funções de monitorização de erros

Apêndice I: Organização da equipa do SGC

Gestão do Projeto

Sérgio Pinto

Equipa Funcional

Equipa Técnica

SAP

UFE

RELATÓRIOS

Ana Luísa
Batista

António
Fryxell

Joana
Calado

Miguel
Perestrelo

Milene
Bação

Sofia Pinho

Paulo Pinto

José Girão

Ruben
Fernandes

Ana Rita
Caeiro

Ana Sofia Mendes

Inês Roque

Apêndice II: SQL dos principais objetos da base de dados

Dada a elevada quantidade de objetos da base de dados, optou-se pela inclusão do SQL referente às dimensões e tabelas de factos que fazem parte dos exemplos apresentados neste relatório.

Tabela de extração de dados de domínio

```
-- TIPO_CLIENTE
CREATE TABLE e_zfv_bpkind (
  bpkind character varying,
  text40 character varying
);
```

Tabela de extração de dados-mestre/dados transacionais

```
CREATE TABLE e_zv_client_rep (
  opbuk CHARACTER VARYING,           -- EMPRESA: código identificador
  name_first CHARACTER VARYING,      -- CLIENTE: 1º nome
  name_last CHARACTER VARYING,       -- CLIENTE: 2º nome
  name_org1 CHARACTER VARYING,       -- EMPRESA: 1º nome
  name_org2 CHARACTER VARYING,       -- EMPRESA: 2º nome
  type_pn CHARACTER VARYING,         -- PARCEIRO: categoria
  bpkind CHARACTER VARYING,          -- CLIENTE: tipo
  partner CHARACTER VARYING,         -- CLIENTE: código identificador
  title CHARACTER VARYING,           -- CLIENTE: título
  xsexm CHARACTER VARYING,           -- CLIENTE: masculino
  xsexf CHARACTER VARYING,          -- CLIENTE: feminino
  natio CHARACTER VARYING,           -- CLIENTE: nacionalidade
  taxnum CHARACTER VARYING,          -- CLIENTE: NIF
  idnumber CHARACTER VARYING,        -- CLIENTE: número identificação
  type_id CHARACTER VARYING,         -- IDENTIFICAÇÃO: tipo
  tel_number CHARACTER VARYING,      -- CLIENTE: número telefone
  smtp_addr CHARACTER VARYING,       -- CLIENTE: e-mail
  addrnumber CHARACTER VARYING,      -- MORADA: código identificador
  cuser_ufe CHARACTER VARYING,       -- USER: código identificador
  cdate CHARACTER VARYING,           -- CLIENTE: data criacao registo
  muser_ufe CHARACTER VARYING,       -- USER: user que fez a ultima modificacao
  mdate CHARACTER VARYING            -- CLIENTE: data modificacao registo
);
```

Tabela de transformação de dados de domínio

```
-- Sequence: dw.seq_client_type
CREATE SEQUENCE dw.seq_client_type
    INCREMENT 1
    MINVALUE 0
    MAXVALUE 9223372036854775807
    START 0
    CACHE 1;
ALTER TABLE dw.seq_client_type
    OWNER TO postgres;

-- Table: dw.dim_client_type
CREATE TABLE dw.dim_client_type(
    id_client_type integer DEFAULT nextval('dw.seq_client_type'::regclass),
    cd_client_type character varying,
    ds_client_type character varying,
    fl_active integer,
    id_load integer,
    dt_update integer
)
WITH (
    OIDS=FALSE
);
ALTER TABLE dw.dim_client_type
    OWNER TO postgres;
```

Tabela de transformação fase I de dados-mestre/dados transacionais

```
-- Table: dw.dim_client
CREATE TABLE dw.dim_client (
    id_client integer DEFAULT nextval('dw.seq_client'::regclass),
    cd_client character varying,
    id_company integer,
    id_partner_category integer,
    first_name_client character varying,
    last_name_client character varying,
    first_name_org character varying,
    last_name_org character varying,
    title character varying,
    gender character varying,
    nationality character varying,
    tax_number character varying,
    id_card_number character varying,
    id_id_type integer,
    id_client_type integer,
    phone_number character varying,
    e_mail character varying,
    dt_creation date,
    dt_modification date,
    id_address integer,
    id_user integer,
    id_regioarea integer,
    fl_active integer,
    id_load integer,
    dt_update integer
)
WITH (
    OIDS=FALSE
);
ALTER TABLE dw.dim_client
    OWNER TO postgres;
```

Tabela de transformação fase II de dados-mestre/dados transacionais

```
CREATE TABLE dw.report_clients (  
  id_client integer,  
  cd_client character varying,  
  id_company integer,  
  id_partner_category integer,  
  title character varying,  
  name character varying,  
  gender character varying,  
  nationality character varying,  
  tax_number character varying,  
  id_card_number character varying,  
  id_id_type integer,  
  id_client_type integer,  
  phone_number character varying,  
  e_mail character varying,  
  dt_creation date,  
  dt_modification date,  
  id_address integer,  
  id_uc integer,  
  id_island integer,  
  id_zone integer,  
  id_region integer,  
  ds_address character varying,  
  postal_code character varying,  
  postal_box character varying,  
  locality character varying,  
  postal_designation character varying,  
  country character varying,  
  id_user integer,  
  id_regioarea integer  
)  
WITH (  
  OIDS=FALSE  
)  
;  
ALTER TABLE dw.report_clients  
  OWNER TO postgres;
```

Tabela de monitorização de erros

```
CREATE TABLE dw.execution_log (  
    execution_id character varying,  
    project character varying(50),  
    job_id character varying(50),  
    job character varying(50),  
    start_time timestamp without time zone,  
    end_time timestamp without time zone,  
    status integer,  
    dsc_status character varying(10),  
    failure_task character varying(100),  
    error_code character varying(1000),  
    job_serial_id bigserial NOT NULL,  
    count_executions smallint,  
    cod_processo character varying(150),  
    dsc_processo character varying(150)  
)  
WITH (  
    OIDS=FALSE  
);  
ALTER TABLE dw.execution_log  
    OWNER TO postgres;
```

View sobre uma tabela de extração de domínio

```
-- View: t_dim_tipo_cliente  
CREATE OR REPLACE VIEW t_dim_tipo_cliente AS  
SELECT '00'::character varying AS id,  
    '-'::character varying AS dsc  
UNION ALL  
SELECT btrim(a.bpkind) AS id,  
    initcap(btrim(a.text40)) AS dsc  
FROM e_zfv_bpkind a  
ORDER BY 1;  
  
ALTER TABLE t_dim_tipo_cliente  
    OWNER TO postgres;  
GRANT ALL ON TABLE t_dim_tipo_cliente TO public;  
GRANT ALL ON TABLE t_dim_tipo_cliente TO postgres;
```

View sobre uma tabela de extração de dados-mestre/dados transacionais

```
-- View: t_dim_cliente
CREATE OR REPLACE VIEW t_dim_cliente AS
SELECT
    btrim(c.opbuk) AS cd_empresa,
    btrim(initcap(c.name_first)) AS primeiro_nome_cliente,
    btrim(initcap(c.name_last)) AS ultimo_nome_cliente,
    btrim(initcap(c.name_org1)) AS primeiro_nome_organizacao,
    btrim(initcap(c.name_org2)) AS segundo_nome_organizacao,
    btrim(c.type_pn) AS cd_categoria_parceiro,
    btrim(c.bpkind) AS cd_tipo_cliente,
    btrim(c.partner) AS cd_cliente,
    CASE
        WHEN c.title::text = '0001'::text THEN 'Sra'::text
        WHEN c.title::text = '0002'::text THEN 'Sr'::text
        WHEN c.title::text = '0003'::text THEN 'Empresa'::text
        ELSE ''::text
    END AS titulo,
    CASE
        WHEN c.xsexm::text = 'X'::text THEN 'M'::text
        WHEN c.xsexf::text = 'X'::text THEN 'F'::text
        ELSE ''::text
    END AS genero,
    btrim(c.natio) AS nacionalidade,
    btrim(c.taxnum) AS nif,
    btrim(c.idnumber) AS n_identificacao,
    btrim(c.type_id) AS cd_tipo_identificacao,
    btrim(c.tel_number) AS n_telefone,
    btrim(c.smtp_addr) AS e_mail,
    btrim(c.addrnumber) AS cd_morada,
    btrim(c.cuser_ufe) AS cd_operador_criador,
    CASE
        WHEN substr(c.cdate::text, 1, 4) = '0000'::text THEN NULL::date
        ELSE btrim(c.cdate)::date
    END AS dt_criacao_registro,
    btrim(c.muser_ufe) AS cd_operador_modificador,
    CASE
        WHEN substr(c.mdate::text, 1, 4) = '0000'::text THEN NULL::date
        ELSE btrim(c.mdate)::date
    END AS dt_modificacao_registro,
    btrim(c.campo_uc) AS cd_unidade_comercial
FROM e_zv_client_rep c;
ALTER TABLE t_dim_cliente
OWNER TO postgres;
GRANT ALL ON TABLE t_dim_cliente TO public;
GRANT ALL ON TABLE t_dim_cliente TO postgres;
```

View sobre uma tabela de transformação de domínio

```
CREATE OR REPLACE VIEW dw.vw_dim_client_type AS
SELECT id_client_type AS id,
       ds_client_type AS dsc
FROM dw.dim_client_type
WHERE fl_active = 1;
```

View sobre uma tabela de transformação de dados-mestre/dados transacionais

```
-- View: dw.vw_report_clients
CREATE OR REPLACE VIEW dw.vw_report_clients AS
SELECT empresa.ds_company AS empresa,
       empresa.id_company AS id_empresa,
       empresa.cd_company AS cd_empresa,
       cliente.cd_client AS n_cliente,
       parceiro.ds_partner_category AS categoria,
       parceiro.id_partner_category AS id_categoria,
       tipocliente.ds_client_type AS tipo_cliente,
       tipocliente.id_client_type AS id_tipo_cliente,
       cliente.name AS nome,
       cliente.title AS titulo,
       cliente.gender AS genero,
       cliente.nationality AS nacionalidade,
       cliente.tax_number AS nif,
       tipoidentificacao.ds_id_type AS tipo_id,
       tipoidentificacao.id_id_type AS id_tipo_id,
       cliente.id_card_number AS n_id,
       cliente.ds_address AS morada,
       cliente.postal_code AS codigo_postal,
       cliente.postal_box AS caixa_postal,
       cliente.locality AS localidade,
       cliente.postal_designation AS designacao_postal,
       regiao.ds_region AS regiao,
       regiao.id_region AS id_regiao,
       cliente.country AS pais,
       cliente.phone_number AS telefone,
       cliente.e_mail AS e_mail,
       cliente.dt_creation AS data_registo,
       operador.cd_user AS operador,
       operador.id_user AS id_operador,
       ilha.ds_island AS ilha,
       ilha.id_island AS id_ilha,
       uc.ds_uc AS uc,
       uc.id_uc AS id_uc,
       zona.ds_zone AS freguesia,
       zona.id_zone AS id_freguesia,
       unidadecomercial.ds_regioarea AS unidadecomercial,
       unidadecomercial.id_regioarea AS id_unidadecomercial
FROM dw.report_clients cliente
LEFT JOIN dw.dim_company empresa
      ON empresa.id_company = cliente.id_company
LEFT JOIN dw.dim_partner_category parceiro
      ON parceiro.id_partner_category = cliente.id_partner_category
LEFT JOIN dw.dim_client_type tipocliente
      ON tipocliente.id_client_type = cliente.id_client_type
LEFT JOIN dw.dim_id_type tipoidentificacao
      ON tipoidentificacao.id_id_type = cliente.id_id_type
LEFT JOIN dw.dim_region regiao
      ON regiao.id_region = cliente.id_region
LEFT JOIN dw.dim_user operador
      ON operador.id_user = cliente.id_user
LEFT JOIN dw.dim_island ilha
      ON ilha.id_island = cliente.id_island
LEFT JOIN dw.dim_uc uc
      ON uc.id_uc = cliente.id_uc
LEFT JOIN dw.dim_zone zona
      ON zona.id_zone = cliente.id_zone
LEFT JOIN dw.dim_regioarea unidadecomercial
      ON unidadecomercial.id_regioarea = cliente.id_regioarea;
ALTER TABLE dw.vw_report_clients
OWNER TO postgres;
```


Apêndice III: SQL das funções de monitorização de erros

Função *on_job_begin*: função responsável por inserir um novo registo na tabela de monitorização de erros.

```
CREATE OR REPLACE FUNCTION dw.on_job_begin (v_exec_id character varying,
                                             v_job_id character varying,
                                             v_project character varying,
                                             v_job character varying,
                                             v_cod_processo character varying,
                                             v_dsc_processo character varying)

RETURNS void AS
$BODY$
begin
    INSERT INTO dw.execution_log (execution_id, job_id,
                                  project, job,
                                  start_time, status,
                                  dsc_status, count_executions,
                                  cod_processo, dsc_processo)

    VALUES (v_exec_id, v_job_id,
              v_project, v_job,
              current_timestamp, 0,
              'Running', 1,
              v_cod_processo, v_dsc_processo);

end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION dw.on_job_begin (character varying, character varying,
                                character varying, character varying,
                                character varying, character varying)

OWNER TO postgres;
```

Função *on_job_end*: função responsável por atualizar a tabela de monitorização de erros quando um *job* termina corretamente.

```
CREATE OR REPLACE FUNCTION dw.on_job_end (v_exec_id character varying,
                                           v_job_id character varying,
                                           v_project character varying,
                                           v_job character varying)

RETURNS void AS
$BODY$
begin
    UPDATE dw.execution_log
    SET
        end_time = current_timestamp,
        status = 1,
        dsc_status = 'Complete'
    WHERE
        execution_id = v_exec_id AND project = v_project
        AND job_id = v_job_id AND job = v_job;

end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION dw.on_job_end (character varying, character varying,
                              character varying, character varying)

OWNER TO postgres;
```

Função *on_job_error*: função que atualiza a tabela de monitorização de erros quando ocorre algum erro num *job*.

```
CREATE OR REPLACE FUNCTION dw.on_job_error (v_exec_id character varying,  
                                             v_job_id character varying,  
                                             v_project character varying,  
                                             v_job character varying,  
                                             v_failure_task character varying,  
                                             v_exception character varying)  
  
    RETURNS void AS  
$BODY$  
    begin  
        UPDATE dw.execution_log  
        SET  
            end_time = current_timestamp,  
            status = 2,  
            dsc_status = 'Error',  
            failure_task = v_failure_task,  
            error_code = v_exception  
        WHERE  
            execution_id = v_exec_id AND project = v_project  
            AND job_id = v_job_id AND job = v_job;  
    end;  
$BODY$  
LANGUAGE plpgsql VOLATILE  
COST 100;  
ALTER FUNCTION dw.on_job_error (character varying, character varying,  
                                character varying, character varying,  
                                character varying, character varying)  
  
OWNER TO postgres;
```